A Quadratic Deformation Model for Representing Facial Expressions

---

A thesis

submitted in partial fulfilment

of the requirements for the Degree

of

Doctor of Philosophy

in the

University of Canterbury

by

Mohammad Obaid

Supervisory Committee

Associate Prof. Dr. R. Mukundan	Supervisor
Prof. Mark Billinghurst	Co-Supervisor
Dr. Mark Sagar	External-Advisor

University of Canterbury

2011

To my parents

Abstract

Techniques for facial expression generation are employed in several applications in computer graphics as well as in the processing of image and video sequences containing faces. Video coding standards such as MPEG-4 support facial expression animation. There are a number of facial expression representations that are application dependent or facial animation standard dependent and most of them require a lot of computational effort. We have developed a completely novel and effective method for representing the primary facial expressions using a model-independent set of deformation parameters (derived using rubber-sheet transformations), which can be easily applied to transform facial feature points. The developed mathematical model captures the necessary non-linear characteristics of deformations of facial muscle regions; producing well-recognizable expressions on images, sketches, and three dimensional models of faces. To show the effectiveness of the method, we developed a variety of novel applications such as facial expression recognition, expression mapping, facial animation and caricature generation.

Table of Contents

List of Tables

List of Figures

Acknowledgments

Throughout the past few years, many people have given me their support and help to achieve this work. I would like to begin this thesis by thanking them all.

Supervisors: Firstly, I am thankful to my supervisor, Associate Prof. Dr. Mukundan for his patience, guidance, encouragement and knowledge that I have learned from him; all of which have helped me to accomplish this thesis and led me through the difficulties in this research. I thank Prof. Mark Billinghurst, my co-supervisor, for his full support and constant encouragement throughout my thesis. He has certainly taken me into another professional level as he helped me establish research collaborations that have extended my way of thinking and research approach. Special thanks also to my external advisor Dr. Mark Sagar, from Weta Digital studios, for his feedback and comments throughout my PhD.

Collaborators: I would like to thank Dr. Roland Goecke at the Australian National University, Canberra, Australia, and Dr. Jason Saragih, Carnegie Mellon University, Pittsburgh, USA, for allowing me to use their software libraries. I am also thankful to Prof. Catherine Pelachaud and her research team at the TelecomParis Tech, Paris, France, for the warm hosting I have received while conducting a research internship at their laboratory.

Friends and colleagues: I am very thankful to all of my colleagues and friends at the HITLab NZ and the Computer Science and Software En-

gineering Department, as they have certainly supported me and helped me throughout this research. I would like to give special thanks to the HITLab NZ manager Ken Beckman and to Dr. Raphael Grasset for their endless assistance whenever needed. Also, special thanks to my friend Christina Dike (a doctor in process) for her encouragement during the long hours at the laboratory.

Karin Fitz: I am very thankful to this very wonderful person, who has kept my motivation up all the time and has giving her endless help over the past few years.

My Family: Above all I would like to thank my parents and family both in New Zealand and in Jordan for their encouragement, understanding and unconditional love which has helped me all throughout my life. Without their help, this thesis would not have become a reality.

This is a limited list and I would like to thank all the people that helped me directly and indirectly throughout my academic endeavors.

Chapter I

Introduction

Recently, with the exponential growth in the power of computer graphics hardware, facial animation has become increasingly important in various application areas such as games, animation, security and multimedia education. The main application areas for facial animation are in the film industry, gaming, and advertising, where the focus is on the modelling and animation of virtual humans and 3D-digital characters. In this context, facial animation has proven to be one of the most challenging aspects of building and animating a virtual character model. The complexity of the human face and the high sensitivity humans have in recognising facial emotions makes representing facial expressions complicated and challenging to produce. This thesis presents a novel approach to representing facial expressions in terms of mathematical transformation functions. The main advantage of our approach is the generic representation of facial expressions that can be employed in facial expression applications such as facial animations and recognition.

This chapter is organised as follows. Section 1.1 presents the goals and the motivation behind our work. Section 1.2 lists the main contributions of this thesis. Section 1.3 outlines all the related publications. Section 1.4 describesthe internships and research visits conducted that relate to this thesis. Section 1.5 gives a summary of the thesis chapters.

1.1 Thesis Goals and Motivation

The goal of this thesis is to investigate, develop and evaluate a generic mathematical representation of facial expressions that is broadly applicable and can be used to synthesize and animate facial models.

Facial animation research has produced a number of facial expression representations that depend on their application semantics or facial animation standards, and most of them are computationally intensive. Developing a truly generic and universal representation of facial expressions is a very difficult problem and is an open research field. This situation has motivated us to research and propose a simple and effective solution for representing facial expressions as a set of model-independent region transformations, offering an easy-to-implement facial expression model. The proposed representations can also support a simple parametric animation of facial expressions using face images or three dimensional mesh models of faces; moreover, it is not only applicable to facial animations but broadly applicable to other applications areas such as expression synthesis, electronic entertainment or affective computing.

The specific goals of this thesis are to:

- Develop a generic model-independent framework for representing all of the primary facial expressions.

- Develop algorithms for mapping facial expressions on two-dimensional images and 3D models.

- Develop algorithms for facial expression recognition from tracked sequences of feature points.

- Develop methods for facial expression animation using the model representations.

- Evaluate the proposed algorithms using experimental analysis and comparative studies.

1.2 Contributions

The main contributions of this thesis are:

- The development of a completely novel and effective generic method for representing all primary facial expressions by capturing the necessary non-linear characteristics of facial muscle deformations using quadratic deformation models (Chapter 3).

- The development of a facial recognition system by using a facial feature tracking system (based on active appearance models) with the quadratic deformation models (Chapter 4).

- The development of new methods for caricature generation and rendering. The novelty comes from being able to manipulate the facial appearance and expressivity of the caricature using real facial muscle deformations (Chapter 5).

- The development of techniques to generate and animate different expressive states of MPEG-4 (Moving Picture Experts Group) compliant 3D virtual agents using the quadratic deformation model representations of facial expressions (Chapter 6).

- Perform extensive experimental analysis to validate the algorithms and to compare their relative performance with respect to existing techniques.

1.3 PhD Publications

Material from this thesis has been previously published in the peer-reviewed papers listed below. The chapters of the thesis that relate to the publications are noted in brackets.

- **M. Obaid**, R. Mukundan, M. Billinghurst, and C. Pelachaud. Expressive MPEG-4 Facial Animation using Quadratic Deformation Models. In Proceedings of the International Conference on Computer Graphics, Imaging and Visualisation, CGIV 2010. IEEE Computer Society. August 9 - 11, 2010, Sydney, Australia. (Chapter 6)

- **M. Obaid**, R. Mukundan, and M. Billinghurst. Generating and Rendering Expressive Caricatures. (Poster) In Proceedings of the International Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2010. 25-29 July, 2010. Los Angeles. (Chapter 5)

- **M. Obaid**, R. Mukundan, and M. Billinghurst. Rendering and Animating Expressive Caricatures. In Proceedings of the Third IEEE International Conference on Computer Science and Information Technology, IEEE ICCSIT2010. 9-11 July 2010, Chengdu, China. (Chapter 5)

- **M. Obaid**, D. Lond, R. Mukundan, and M. Billinghurst. Facial Cari-

cature Generation using a Quadratic Deformation Model. In Proceedings of the International Conference on Advances in Computer Entertainment Technology, ACE 2009. 29-31 October 2009, Athens, Greece. (Chapter 5)

- **M. Obaid**, R. Mukundan, R. Goecke, M. Billinghurst, and H. Seichter. A Quadratic Deformation Model for Facial Expression Recognition. Digital Image Computing and Applications, DICTA 2009. 1-3 December 2009, Melbourne, Australia. (Chapter 4)

- **M. Obaid**, R. Mukundan, M. Billinghurst, and M. Sagar. Facial Expression Representation using a Quadratic Deformation Model. In Proceedings of the International Conference on Computer Graphics, Imaging and Visualisation, CGIV 2009. IEEE Computer Society. August 11 - 14, 2009, Tianjin, China. (Chapter 3)

I co-authored the following paper as part of a final year engineering project at the University of Canterbury.

- C. Tay, **M. Obaid**, R. Mukundan, and A. Bainbridge-Smith. Facial Expressions using a Quadratic Deformation Model: Analysis and Synthesis. Image and Vision Computing New Zealand, IVCNZ 2009. 23-25 November 2009. (Part of this paper is covered in Section 3.6)

During my PhD, and as part of the literature review, I published the following.

- **M. Obaid**, C. Han, and M. Billinghurst. "Feed The Fish": An Affect-Aware Game. In Proceedings of the 5th Australasian Conference on

Interactive Entertainment 2008, IE 2008. 3-4 December 2008, Brisbane, Australia.

1.4 Internships/Research Visits

- February 2009. Research visit to the School of Computer Science at the Australian National University (ANU), Canberra, Australia. Recieved comprehensive training on how to use and apply the the active appearance model markerless tracking system.

- September-December 2009. Internship at the CNRS LTCI, TELECOM ParisTech Institute, Paris, France. Worked with the Greta conversational virtual character and motion captured data (mocap).

1.5 Chapter Summary

This section describes the chapters of this thesis.

Chapter 2 - Background: reviews the background material and research literature used to derive the mathematical representations of facial expressions. An overview of the facial muscles' anatomy, followed by a description of the six universal facial expressions (surprise, fear, anger, sadness, disgust, and smile) is given.

A review of facial animations and their control techniques is described with a focus on the widely used standards: Facial Action Coding System (FACS) and MPEG-4 Facial Animation Standard. The literature of the commonly used facial deformation models to animate 3D faces is also described in this chapter.

Finally, work related to facial application areas are outlined with a focus on the following applications: facial expression recognition, facial caricature generation, and MPEG-4 facial animation.

Chapter 3 - Facial Expression Representation Using Quadratic Deformation Models: conveys the main contribution within this thesis by describing a novel approach for mathematically representing facial expressions. A generic mathematical representation of facial expressions is described by modelling the non-linear nature of muscle deformations into rubber-sheet transformations (or quadratic deformation models).

Chapter 4 - Facial Expression Recognition using Quadratic Deformation Models: describes a new approach for classifying facial expressions by using a facial feature tracking system with the quadratic deformation models. An evaluation of the recognition system and the facial expression databases used are reported in this chapter.

Chapter 5 - Expressive Caricatures using Quadratic Deformation Models: presents a novel approach for generating expressive caricatures from a given face image, with the ability to map any one of the six main expressions and control the degree of its expressiveness on the generated caricature. A stroke based painterly rendering algorithm is also described to generate a more stylized caricature. A comparative analysis of the different facial caricature algorithms is reported in this chapter.

Chapter 6 - Facial Animation using Quadratic Deformation Models: describes how quadratic deformation models are used to synthesize facial expressions on 3D faces. It is demonstrated by applying the quadratic deformation models on a 3D face compliant with the MPEG-4 facial animation standard.

Chapter 7 - Conclusion and Future Work: provides a summary of the contributions of this thesis, limitations of the work presented and proposes future research directions.

Chapter II

Background

2.1 Introduction

Research on mathematically representing facial expressions originates as a sub-area of the character animation techniques. This chapter will review some of the related background research used to derive and validate the mathematical representation of facial expressions. This chapter is organised as follows. Section 2.2 gives a brief overview of the facial muscle anatomy. Section 2.3 describes the six main universal facial expressions. Section 2.4 provides details of the most commonly used facial parameterization standards (Facial Action Coding System (FACS) and MPEG-4 Facial Animation standard). Section 2.5 presents the commonly used facial deformation model techniques. Section 2.6 outlines related work of different facial expression application areas.

2.2 Anatomy of the Face Muscles

Facial expressions are the result of the large number of facial muscle layers located between the skull and the skin. Combining the different muscle movements allows a large number of different facial expressions to be gener-

ated.

There are more than 50 muscles located around the skull, which are classified as singular or groups of smaller muscles [77]. They can be described as the muscles of facial expression and they act like elastic rubber sheets that are stretched over facial bones and organs [22]. Generally, facial muscle motion is associated with its attachment points *insertion* and *origin*. The insertion point is the movable end of the muscles and it is generally attached to the skin, while the origin point is the fixed end of the muscles and it is attached to the bone. The facial nerves innervate muscles which in turn controls their activity, as singular or groups of muscles, to form facial expressions [77]. The following section describes in detail the muscles of facial expression and their primary actions.

2.2.1 Muscles of Facial Expression

Figures 2.1 illustrates the frontal view of the facial muscle anatomy of the outermost layer [77]. The illustrated facial expression muscles do not work independently but work as a team of muscles. They can be grouped into an upper and lower face muscles and their definition is based on the orientation of their individual muscle fibers. Three types of orientations are described according to how muscles are pulled at their attachment points [77]. Generally, the three types of muscle motion are defined as [105]:

- **Parallel/Linear**: muscles that pull in an angular direction.
- **Circular/Spiralized**: sphincter muscles that squeeze.
- **Sheet**: a group of linear muscles that cover an area.

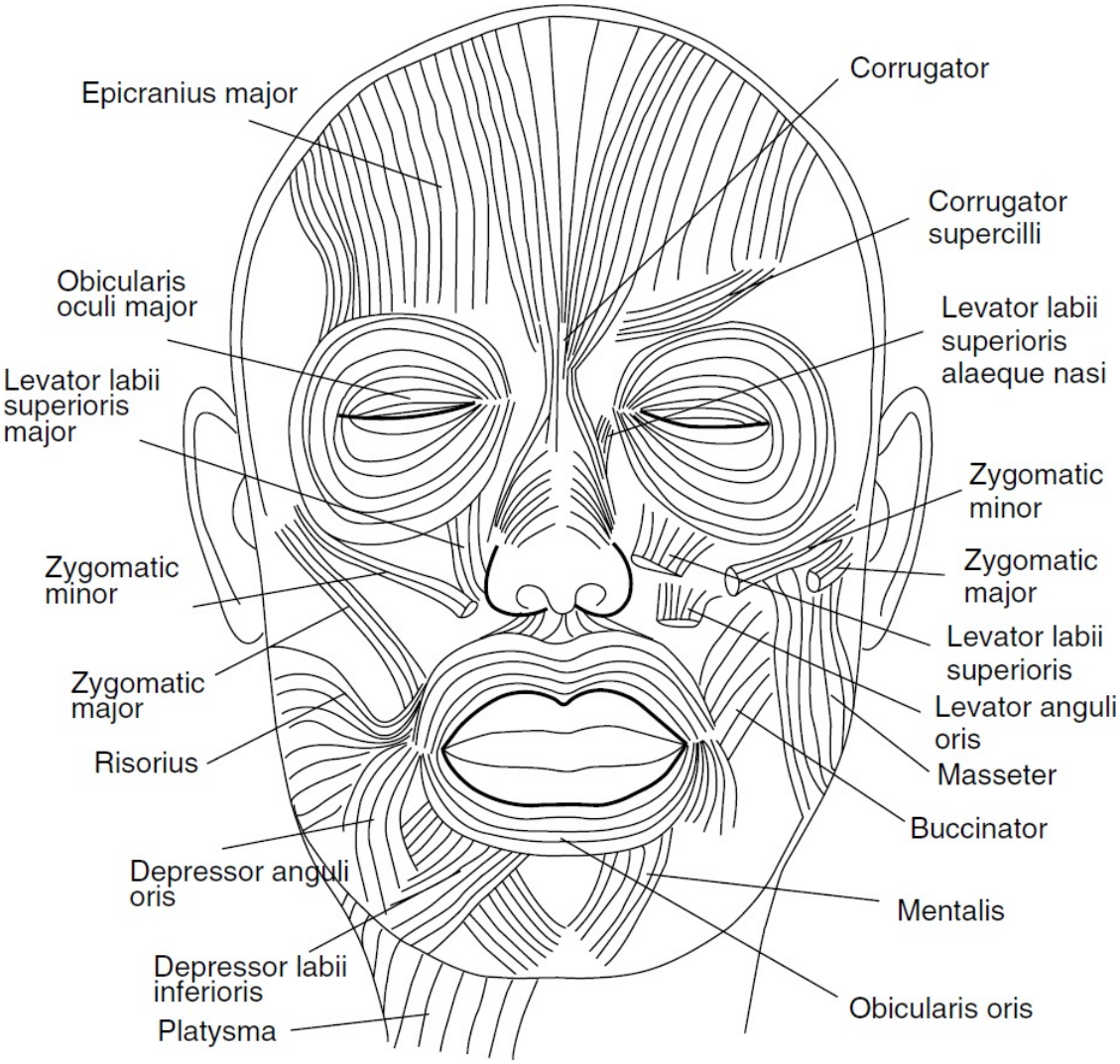


Figure 2.1: Anatomy of Facial Muscles. Image obtained from [77].

The following list gives a brief description of the basis of the facial expression muscles, the group they belong to, and their primary actions[1].

- **Epicranius Major (Frontalis)**: A smooth thin muscle that is slightly curved and in a quadrilateral shape.

  Action: The muscle allows for the full expansion and contraction of the eyebrows. It also moves the scalp backwards and forwards.

  Group: Scalp

- **Orbicularis Oculi**: A circular muscle around each eye.

  Action: The muscle allows for the opening and closing of the eye(s).

  Group: Eye

- **Zygomaticus Major**: An extended and narrow muscle that originates from the ear and reaches across the face to the corner of the mouth.

  Action: The muscle elevates upward and outward, as in a smile.

  Group: Mouth

- **Zygomaticus Minor**: Similar to the Zygomaticus Major muscle but it reaches across the face to the upper lip.

  Action: The muscle elevates the upper lip upwards.

  Group: Mouth

- **Platysma**: A muscle positioned between the collar bone and the lower mouth region.

  Action: The muscle pulls the lower lip downward and backward.

  Group: Neck

[1] The described facial muscles are based on a more detailed description of the facial muscle anatomy which can be found in [77] [38] [45].

- **Corrugator {supercilii}**: A small straight muscle located at the inner part of the eyebrow.

  Action: The muscle extends the eyebrows towards the middle and downwards.

  Group: Eye

- **Nasalis**: The muscle is located on either side of the nose. The muscle has two parts, the compressor nasalis and the dilator nasalis.

  Action: When the muscle tightens it opens/widens the nostril and raises the upper lip.

  Group: Nose

- **Levator labii superioris alaeque nasi muscle**: The muscle starts at the upper part of the frontal process of the maxilla and extends to the upper lip.

  Action: When the muscle tightens it elevates the corners of the nostrils.

  Group: Nose

- **Levator labii superioris {major}**: Originates just under the lower eyelid and extends across nose bone to the upper lip.

  Action: The muscle raises the upper lip and the nose sides.

  Group: Mouth

- **Lavator anguli oris (Caninus)**: This muscle is a deep muscle that originates from the upper part of the Canine Fossa and extends across to the upper lip.

  Action: Raises the upper lip.

Group: Mouth

- **Masseter**: The muscle is situated in the cheek area on each side of the face.

  Action: Elevates the mandible and also retracts the mandible.

  Group: Muscles of Mastication

- **Buccinator**: The muscle is situated in the cheeks.

  Action: When contracted the muscle closes the mouth and compresses the cheek.

  Group: Mouth

- **Risorius**: The muscle is situated in the cheek area on both sides of the mouth.

  Action: The muscle acts in a similar way to the Buccinator.

  Group: Mouth

- **Orbicularis Oris**: Is a muscle that orbits the mouth.

  Action: The muscle closes and compresses the lips.

  Group: Mouth

- **Depressor Labii Inferioris**: A quadrilateral muscle situated on both sides of the face between the lower edge of the jaw and the lower lip.

  Action: The muscle moves the lower lip upwards and downwards.

  Group: Mouth

- **Depressor Anguli Oris**: A triangular muscle situated on both sides of the face in the lower-jaw area.

Action: The muscle moves the corners of the mouth upwards and downwards.

Group: Mouth

- **Mentalis**: Situated on both sides of the centre face of the chin.

  Action: Elevates the chin.

  Group: Mouth

2.3 Facial Expressions

"A facial expression results from one or more motions or positions of the muscles of the face. These movements convey the emotional state of the individual to observers. Facial expressions are a form of nonverbal communication. They are a primary means of conveying social information among humans, but also occur in most other mammals and some other animal species" [110].

Researchers have debated the exact number of the primary basic facial expressions. Charles Darwin's book in 1872, titled 'The Expression of the Emotions in Man and Animals', is recognised as the first published work to define the basic facial expressions. He classified all expressions into 13 categories. More scientific research concludes that facial expressions consist of six universal cross-cultural categories [29]. Paul Ekman (who is one of the founders of the Facial Action Coding System (FACS) described in Section 2.4.2) groups the universal facial expressions into the following six categories (as shown in Figure 2.2): surprise, fear, anger, sadness, disgust, and smile. Each of these categories has a number of intermediate facial expressions that are based on the intensity level and the expression details (we explain intermediate expressions in Section 2.3.7).

Figure 2.2: The six universal facial expression categories described by Paul Ekman. Images obtained from [29].

The following sections describe the universal facial expressions and outline what portion of the face allows for the facial expression appearance. The information presented in the following sections is based on the work presented in [77] [38].

2.3.1 Smile

The 'smile' facial expression corresponds to the internal feeling or the emotional state of happiness, pleasure, love or joy. A smile can vary in how it is expressed by humans depending on their emotional feelings; this variation includes a simple smile, an open mouth smile, or laughter. The smiling facial expression can also be false, such as a false smile or false laughter [77].

A simple smile is a combination of raising the corners of the mouth up, a slight tightening to the eyelids, a raise to the cheeks, and the relaxation of the eyebrows.
The muscles involved are the Lavator labii superioris, Risorius, Zygomaticus major, and Zygomaticus minor. Figure 2.2 (a) illustrates the 'smile' facial expression.

2.3.2 Fear

The 'fear' facial expression is a response to being under threat. Fear can vary from worry to terror [77] depending on the situation that the individual is in, such as pain or danger.

Expressing fear is a combination of bringing the eyebrows together, raising the eyelids and stretching the mouth corners horizontally. Figure 2.2 (b) illustrates the 'fear' expression.

The muscles involved are the Frontalis, Orbicularis oculi, Buccinator, Mentalis and Platysma.

2.3.3 Disgust

The 'disgust' facial expression is associated with the internal emotional response to something revolting or repellent [110].

The primary actions for the disgust facial expression are the wrinkling of the nose, the slight closure and lowering of the eyebrows, and the raising of the upper lip while the lower lip is extended.
The muscles involved are the Frontalis, Nasalis, Lavator labii superioris, Lavator anguli oris, Zygomaticus major and Zygomaticus minor. Figure 2.2 (c) illustrates the disgust facial expression.

2.3.4 Surprise

The facial expression of surprise is a result of a short lasting emotional state due to an unexpected event [110].

Surprise is expressed by raising the eyebrows and the eyelids up, while the jaw drops and the mouth is open. Figure 2.2 (d) illustrates the surprise facial expression.
The muscles involved are the Frontalis, Orbicularis oculi, Buccinator, Mentalis and Platysma.

2.3.5 Sad

A sad facial expression is associated with the internal emotional feeling of unhappiness, disadvantage, and negativity. The emotional state can be re-

ferred to as sadness and has many variations including crying, and the look of a negative mood.

In sadness, the inner portions of the eyebrows rise up, the eyelids slightly close and the mouth lip corners drop. Figure 2.2 (e) illustrates the sad facial expression.

The muscles involved are the Frontalis, Corrugator, Nasalis, Depressor labii inferioris, Depressor anguli oris, Mentalis and Platysma.

2.3.6 Anger

The expression of anger is associated with the internal emotional feeling of being offended. Anger has many variations including hostility, being furious and rage.

In anger, the eyelids slightly close, the inner portions of the eyebrows lower and come closer together. In extreme anger, the upper eyelid opens, and the lips compress together. Figure 2.2 (f) illustrates the facial expression of anger.

The muscles involved are the Corrugator, Obcularis oculi major, Levator labii superioris alaeque nasi muscle, Depressor labii inferioris, Depressor anguli oris, Mentalis and Platysma.

2.3.7 Intermediate Expressions

The six universal basic emotional expressions proposed by Paul Ekman [29] have dominated computer science literature. In psychology, researchers explore a wider range of emotional states; however, most research results are not applicable to and cannot be used in the computer science fields [64] [83].

Results are usually verbal descriptions and do not provide a reference that computer scientist can reflect on [64]. Nevertheless, a few studies offer outcomes that can be used in computer science, such as the work presented by Whissel [109].

From psychology, Whissel [109] studied how different activation levels of an expression can form other facial expressions or emotions. In her studies she defined different categories of a facial expression E (from the six universal expression) in two-dimensional space (Activation (vertical axis) and Evaluation (horizontal axis)). Activation refers to the level of arousal with respect to E, while Evaluation refers to the level of agreement [64]. Using a two dimensional emotional space to define expressions provides an appropriate way to translate emotional words to coordinate locations in the emotional space [83].

From the synthetic point of view, evaluation levels can be hard to synthesize as they correspond with the internal emotional of feeling and can be described as a positive or a negative valence. However, activation levels refer to the degree of action presented by the person. This allows us to translate the different activation levels of an expression to intermediate emotional states.

2.4 Control Parameterization

2.4.1 Facial Animation

The facial animations control of a 3D geometric representation of a head model can be carried out by the direct manipulation approach. This is where the geometry parameters of the 3D model, such as vertex points in a polygon

mesh, are manipulated directly by an animator [75]. This approach allows animators to have maximum control; however, knowledge about the human facial structure is required for the direct manipulation approach, which can significantly increase the workload for animators and unskilled users. This process can be made simpler by using another facial animation control approach, which is motion capture. This involves deriving parameters of the face motion from a real individual and then transmitting the captured parameters (control parameters) to a 3D model to construct new facial expressions [15] [9]. Parke [76] noted that it is valuable for an animator to have parameterized models to control animations, one will only have to manipulate a small amount of data to achieve the desired animation. Another advantage of parameterized models is the efficient transmission of the small amount of data representing the expression parameters over communication channels [76].

In the following sections the two widely used parameterization standards for facial animations are described.

2.4.2 Facial Action Coding System (FACS)

The Facial Action Coding System (FACS), originally developed by Paul Ekman and Wallace Friesen [30], is a system that describes facial movements as groups of muscle actions. Their muscular action description is based on the analysis of face behaviour, in which every facial appearance is a result of an individual muscle movement or a small group of muscle movements. Ekman and Friesen defined those muscular actions as Action Units (AU). The initial version of FACS defined forty six AU; more recently, Paul Ekman, Wallace

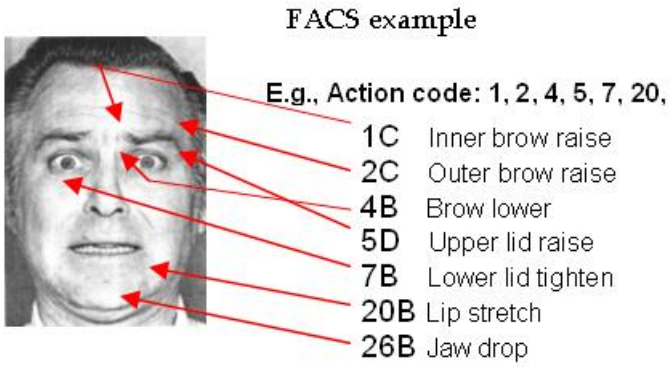


Figure 2.3: An example of the Facial Action Coding System [29].

Friesen, and Joseph Hager [31] revised their description into seventy two AUs (Figure 2.3 shows an example of the facial expression fear coded using FACS AU). FACS is considered to be the most commonly used standard that describes facial appearances. Originally FACS was used by psychologists and behavioural scientists; however, the system is now used in a wide range of different domain areas such as computer animation, computer vision and teleconferencing.

Even with the wide use of FACS, researchers have identified several limitations [33][34]. One of the main limitations, of particular concern to the computer science community, is the lack of representation of the spatiotemporal dynamics of the facial muscle behaviour. As [33] points out, "The lack of temporal and detailed spatial (both local and global) information is a severe limitation of the FACS model [....]. Additionally, the spatial arrangement of facial features also suggests the importance of understanding the face as a mobile, bumpy surface rather than a static flat pattern". Moreover, FACS is a verbal description of muscle movements and does not provide a control mechanism to animate facial expressions.

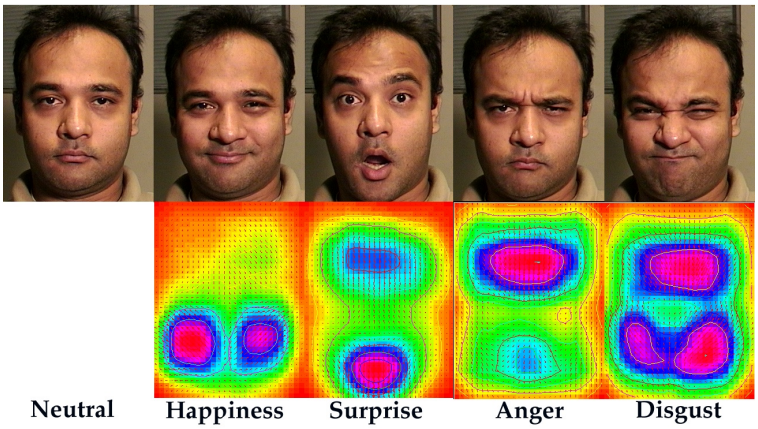


Figure 2.4: Computer vision optical flow technique presented by Essa [34]. Image adopted with permission from the author [32].

FACS++

Essa and Pentland [34] addressed the emotional spatiotemporal limitation of FACS. They introduced a method called FACS++, which uses a computer vision optical flow technique to observe facial spatial information over time. FACS++ uses the observed spatial information to actively estimate and update a dynamic facial model. Figure 2.4 demonstrates the optical flow computation for a video sequence of different facial expressions.

2.4.3 MPEG-4 Facial Animation Standard

The Moving Pictures Experts Group (MPEG) developed the first standardised facial animation (FA) parameterization framework as part of the MPEG-4 standard. The MPEG-4 facial animation standard [72] is defined by 84 feature points (FPs) positioned around the neutral face. FPs are grouped into facial regions and numbered according to their corresponding group. Figure 2.5 shows the MPEG-4 FPs groupings. From the FPs set, the stan-

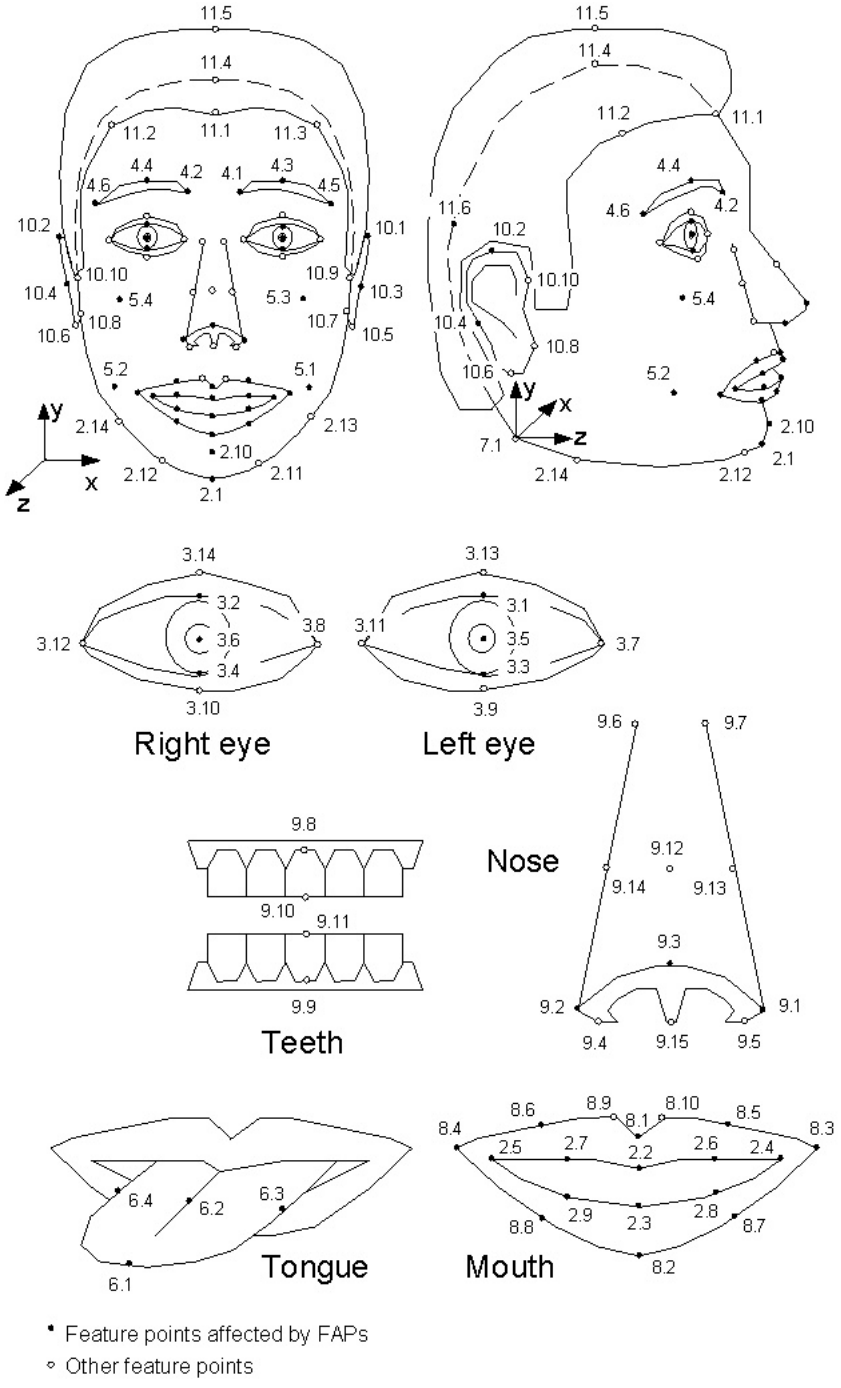


Figure 2.5: MPEG-4 feature points. Image obtained from [98].

Table 2.1: FAP groups [72].

Group/s	**Number of FAPS**
Visemes and expression	2
Jaw, chin, inner lower lip, corner lips, mid lip	16
Eyeballs, pupils, eyelids	12
Eyebrow	8
Cheeks	4
Tongue	5
Head rotation	3
Outer lip positions	10
Nose	4
Ears	4

dard defines a set of 68 Facial Animation Parameters (FAPs) that directly correspond to the FACS AU (described in Section 2.4.2). FAPs consist of two high-level parameters, which specify visemes and expressions, and 66 low-level parameters that describe the movements of feature points defined around the different facial regions (as shown in Table 2.1). Table 2.2 shows an example of the FAP description of the facial actions. The description includes the definition of FAP as a translation magnitude of feature points along one of the three Cartesian axes, while the FAP groups of the head and eyeballs are defined in terms of rotational angle [98].

The facial structure and texture are controlled by the encoding, and decoding of the MPEG-4 FAPS and the Facial Definition Parameters (FDP). FDP play an important role in allowing for the generic employment of FAPs to any face model. FDP contains information that defines the appearance of the 3D face model such as the static geometry, surface properties, and animation rules [98].

Table 2.2: FAPs description [72].

#	FAP Name	FAP Description	Units	Uni /Bidir	Pos Motion	Grp	FDP Sub Grp Num
...	...	...	...	...	...	...	...
3	open_jaw	Vertical jaw displacement does not affect mouth opening	MNS	U	down	2	1
4	lower_t_midlip	Vertical top middle inner lip displacement	MNS	B	down	2	2
5	raise_b_midlip	Vertical bottom middle inner lip displacement	MNS	B	up	2	3
6	stretch_l_cornerlip	Horizontal displacement of left inner lip corner	MW	B	left	2	4
7	stretch_r_cornerlip	Horizontal displacement of right inner lip corner	MW	B	right	2	5
8	lower_t_lip_lm	Vertical displacement of midpoint between left corner and middle of top inner lip	MNS	B	down	2	6
...	...	...	...	...	...	...	...

MPEG-4 FAPs and AUs provide a general description of facial muscle movements, but one common limitation is the lack of defining spatial information of facial deformations [46]. Another limitation of the MPEG-4 facial animation standard is that it does not support a holistic definition and manipulation mechanism for facial expressions and can be considered a low level parameterization standard. In general, an upper layer definition is required on the top of the MPEG-4 standard, to allow for the ease of synthesizing, controlling and manipulating facial animations using FAPs.

2.4.4 Virtual Characters Compliant with MPEG-4

In recent years, several facial animation techniques have been proposed to synthesize and generate facial expressions for 3D virtual characters such as the description of the different methods found in [77] and [82]. Moreover, since the establishment of the MPEG-4 facial animation standard, several 3D synthetic virtual characters and Embodied Conversational Agents (ECA) have emerged to support the MPEG-4 standard [70][71][6]. One of the most well known engines developed based on the MPEG-4 standard is the Greta ECA engine [79]. The design of Greta's face is supported by a 3D facial model compliant with the MPEG-4 facial animation standard and it is capable of simulating, in a believable manner, the dynamic aspects of the human face. In the work presented in this thesis, the Greta facial animation engine is used to generate expressive appearances of Greta's 3D facial model. Figure 2.6 shows an example of Greta's face.

Figure 2.6: An example of Greta's face generated by the Greta Facial Animation Engine [78].

Parameterized Facial Expression Synthesis based on the MPEG-4 Standard

Several approaches have been proposed to synthesize and animate facial expressions based on the MPEG-4 facial animation standard, including the work by Raouzaiou *et al.* [84], who proposed a framework to synthesize and analyze intermediate and primary expressions by creating visual profiles based on MPEG-4 FAPs. Their main focus was to synthesize and animate facial expressions on a 3D facial model compliant with the MPEG-4 standard. Raouzaiou *et al.* defined the six main universal expressions (surprise, fear, anger, sadness, disgust, and smile) using MPEG-4 FAPs.

Raouzaiou *et al.* [84] measured the range of variation for each FAP by analysing automatically tracked facial points of a facial expression database. They measured the distances of the tracked facial feature points and applied statistical analysis to define meaningful measures, such as mean values and standard deviations, for each of the primary facial expressions. Using

the computed statistical measures they defined profiles for the six universal expressions using the FAPs as follows:

- Let $m_{e,j}$ and $\sigma_{e,j}$ be the mean and standard deviation of FAP F_j for the facial expression e

- The allowable variation of expression e is defined as $X_{e,j} = [m_{e,j} - \sigma_{e,j}, m_{e,j} + \sigma_{e,j}]$

Using Whissel's emotional study [109] (described in Section 2.3.7), they generated intermediate expressions based on the computed statistical measurements they obtained. They defined the following rules to compute the intermediate expressions based on Whissel's emotional theory.

- Let P_e^k be the k^{th} profile for expression e

- P_A^k and P_I^k employ the same FAPs, where A is a universal expression and I is the intermediate expression.

- The range of variation is computed by $X_{I,j}^k = (\frac{a_I}{a_A})X_{A,j}^k$, where a_I and a_A are the activation values from the emotion words A and I defined by Whissel's study [109].

2.5 Facial Deformation Models

Animating 3D face models has remained an active field of research since the pioneering work of Parke in 1972 [75]. Since this work, many techniques and approaches have been researched and developed in the area of 3D face modelling and animation. Noh and Neumann [68] categorised previous work

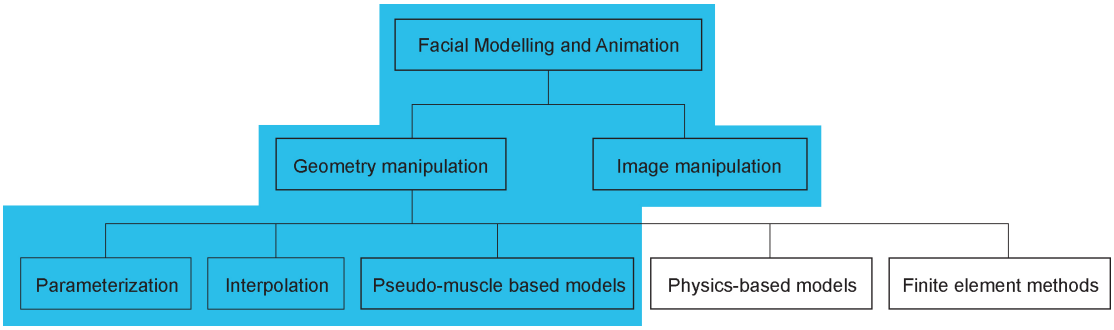


Figure 2.7: Facial modelling and animation methods [68]. The scope of this thesis is shown in the highlighted part of this diagram.

into two groups as shown in Figure 2.7; image-based and geometry-based. Geometry-based techniques refer to the control of 3D face models by deforming the vertices of the model to form the desired 3D representations. On the other hand, image-based techniques morph a combination of 2D-images to achieve the animation effects. Some techniques use a performance-based control method to perform animations by applying captured data of real people to either geometry-based models or image-based models [15] [58].

In this section, we focus on describing related work for geometry based facial deformation techniques as it serves the background literature of our research. A thorough literature review of the methods presented in this section can be found in the following publications [68], Chapter 6 of [54], [77], [24] and [82].

2.5.1 Interpolation

Shape Interpolation is one of the most widely used mechanisms for facial animations. For facial expression animations, the idea is to capture time stamped key-frames between the neutral (E_0) expression and the desired

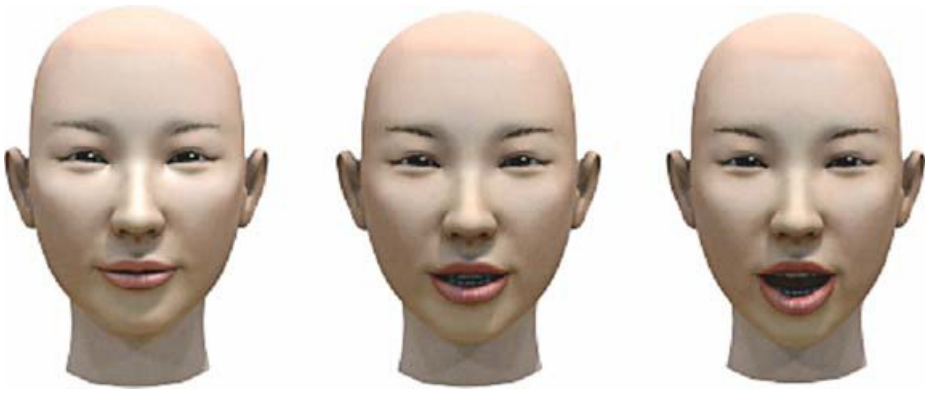


Figure 2.8: Linear interpolation example performed on blend-shapes from a neutral pose to an open mouth pose. Image obtained from [24].

expression (E_1). Interpolation can be represented by the following linear parametric equation:

$$E(t) = (1 - t)E_0 + tE_1 \qquad 0.0 < t < 1.0 \tag{2.1}$$

Animators can generate a facial animation by changing the parameter value t to get the intermediate frames between the neutral and desired expression (as in the example given in Figure 2.8). Linear interpolations are commonly used as they are the simplest [54]; however, other interpolation functions (cosine [106] or spline interpolations) can provide better animation effects as they smooth the start and the end of the animation. Bilinear interpolations are also used to generate a wider range of facial expressions when more than two key-frames are involved [54] [4]. Parke [75] was one of the first people to use shape interpolations in facial animations.

Blend-shapes are a form of surface shape interpolation [77]. In recent years, blending techniques seem to be of more interest to researchers due

to the fact they are widely used in the film making industry. For example, according to Lewis *et al.* [60], a total of 946 blend-shapes are used in the Gollum model in Lord of the Rings movies.

Recent developments include the work of Deng *et al.* [23]. They presented a semi-automatic technique to directly animate popular 3D blend-shape face models by mapping facial motion capture data spaces to 3D blend-shape weights [23]. Joshi *et al.* [51] proposed a method which automatically performs blend-shape segmentations into smaller regions. This allows the capturing of complex facial expressions with a great reduction of manual work.

Shape interpolations have several limitations that include: a restriction on the size and range of facial expressions, the need for large amounts of data for each expression (such as a range of different expression poses), and independent face animations that are hard to generate.

2.5.2 Parametric Models

To overcome the limitations of the shape interpolation techniques, Parke [76] introduced a parametric model to control the facial appearance of the face polygons. An example of the 3D face model is shown in Figure 2.9. Using a parameterized face model allows the animator to control different regions of the face using a small set of parameters that can generate a larger range of facial animations. The limitations of this technique are a dependency on the topology and structure of the face, which in turn limits the generality of the technique. It is also difficult to obtain a complete set of parameters that allow for a full range of facial expressions.

Figure 2.9: The parameterized model of Parke [77].

2.5.3 Muscle-based Models

The anatomy of the face is very complex and it is very difficult to generate a generic model that addresses all aspects of the face's anatomy. To achieve a simplified model that is generic, researchers have looked at modelling the characteristics of facial muscles, bones, and skin. This approach allows for the portability of the model as facial muscles are the same for all humans. Two main techniques have been proposed by the research community: (1) Physics-based modelling and (2) Pseudo-muscle modelling.

Physics-based Models:

Physics-based modelling is a form of representing the underlying physics and mechanism of the facial muscles. This type of modelling is categorised into three types: (1) spring mesh muscle, (2) layered spring mesh muscle and (3) vector muscle.

Spring mesh muscle represents the elasticity of the muscles using the physics of spring forces. Platt and Badler [80] were some of the first people

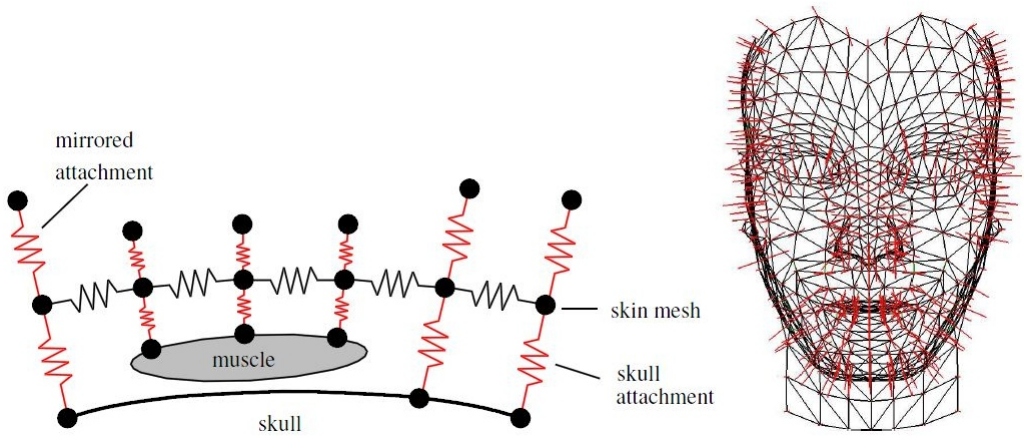


Figure 2.10: Spring Mesh system presented by Kaehler [52].

to attempt modelling of the facial skin and muscles as elastic spring forces. In their work they defined the facial skin layer as a mesh of interconnected springs. Springs are also defined to connect the skin layer with the underlying muscle layer. This setup allows the facial muscles to control the deformation of the skin based on the defined spring forces.

Further work in this field advanced to represent the different facial layers (skin, fatty tissues and muscles) as layers of interconnected springs [100] [59]. Lucero and Munhall [63] extended this work to take into account the actual biomedical properties of facial skin and muscles. More recently, Kaehler [53] introduced a tool to specify physics based spring muscles into 3D face geometry. An example of the spring mesh system presented by Kaehler is shown in Figure 2.10.

Waters [105] derived from linear, sheet and sphincter muscles a system that uses vectors and radial functions to deform a skin mesh. He demonstrated his model by animating a 3D face model to convey the six main facial expressions (surprise, fear, anger, sadness, disgust, and smile). Wa-

ters' work still serves as the basis of most physics-based facial animation techniques used today.

Pseudo-Muscle Based Models:

Pseudo-muscle modelling works by simulating the muscle's behaviour without taking into account the dynamics of the underlying muscular anatomy. Muscles are simulated in the form of splines or free form deformation.

Spline pseudo muscles are introduced to support smooth and flexible deformations over polygonal models that fail to represent smoothness and flexibility when animated [54]. Many techniques presented by researchers make use of splines such as the work presented in [103] [39] [28].

Free form deformation (FFD) is a techniques which deforms 3D objects based on the manipulation of control points arranged in a 3D lattice [54]. Barr [7] presented a method that allows for changing deformations while the 3D object is transforming. Sederberg and Parry [90] presented a more generalised method to define and control the deformations of an object within an embedded control box. An extension to this work is Rational Free Form Deformation (RFED) [55], which incorporates weight factors to allow for more flexibility in deformation control [54]. One of the main advantages of using FED or RFED is that deformations do not depend on the characteristics of the surface itself. However, this limits the realistic simulation of the facial muscles and skin behaviour as it does not provide a precise simulation of the actual muscle and skin behavior since it is based on surface volumetric deformations and it does not account for the volumetric changes occurring in the physical muscle [24] [54].

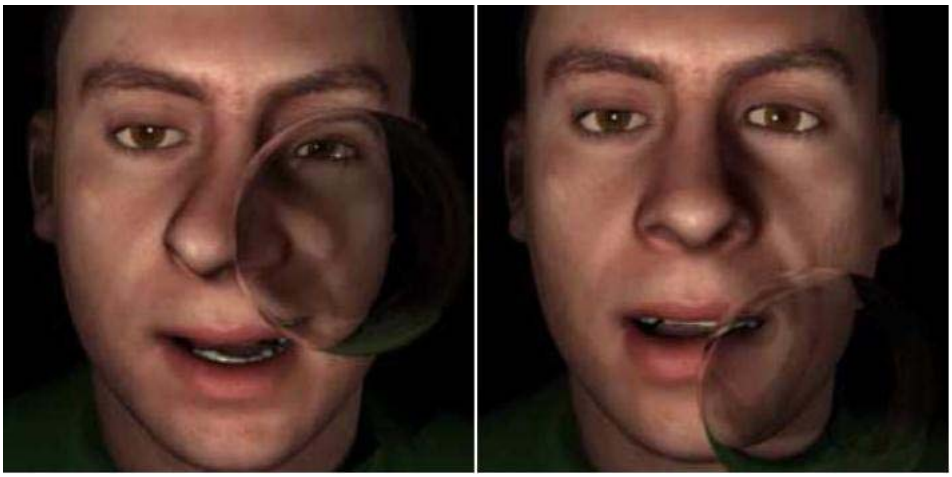


Figure 2.11: An example of FEM and the effects of a colliding force on a face model (Sifakis *et al.* [93]).

2.5.4 Finite Element Method

Finite Element Methods (FEM) are used widely in structural analysis of materials in CAD/CAM applications [54]. FEM work by approximating the solution of a continuous function with a series of shape functions. For their accuracy, FEM are employed in biomechanical and medical simulations [54]. Recently, finite element algorithms have been used by Sifakis *et al.* [93] to automatically build an accurate muscle-based facial model based on the muscle construction principles of Teran *et al.* [99]. Sifakis *et al.* [93] also introduced the novel feature of the facial muscles being able to interact with external objects, such as a collision or contact force (Figure 2.11).

2.6 Application Areas

The analysis and synthesis of facial expressions are important in various application areas, such as games, cartoon animation, teleconferencing, security,

and multimedia. Moreover, most application areas refer to *synthesis* as generating expressive 3D-digital characters and refer to *analysis* as a way of understanding and perceiving the user's facial expression.

In this section, related work in a range of application areas that use facial expression synthesis and analysis is described. Specifically, the related body of work in the area of facial expression recognition is first introduced, and then a literature overview of facial caricature generation and rendering is given. Finally, the related work background on animating 3D expressive characters compliant with the MPEG-4 facial animation standard is described.

2.6.1 Facial Expression Recognition

Recognition of facial expressions with computer vision based methods has been used in many application areas, such as ambient interactivity in multimodal and affective applications, or e-learning and security software where salient expression changes can be used to guide interaction strategies.

For the past two decades, facial expression recognition has attracted many computer vision researchers. Figure 2.12 shows a generic framework used to analyse and classify facial expressions. In general, there are two main approaches to facial expression recognition and tracking: (1) holistic (processes the face as a whole) and (2) feature based (focuses on facial feature patterns) [36, 114]. A number of methods, representing both approaches, have been proposed by the computer vision community for the task of facial feature tracking and recognition including Principal Component Analysis (PCA) [12], Gabor Wavelets [21], hybrid approaches [10], and more recently Active Appearance Models (AAM) [25].

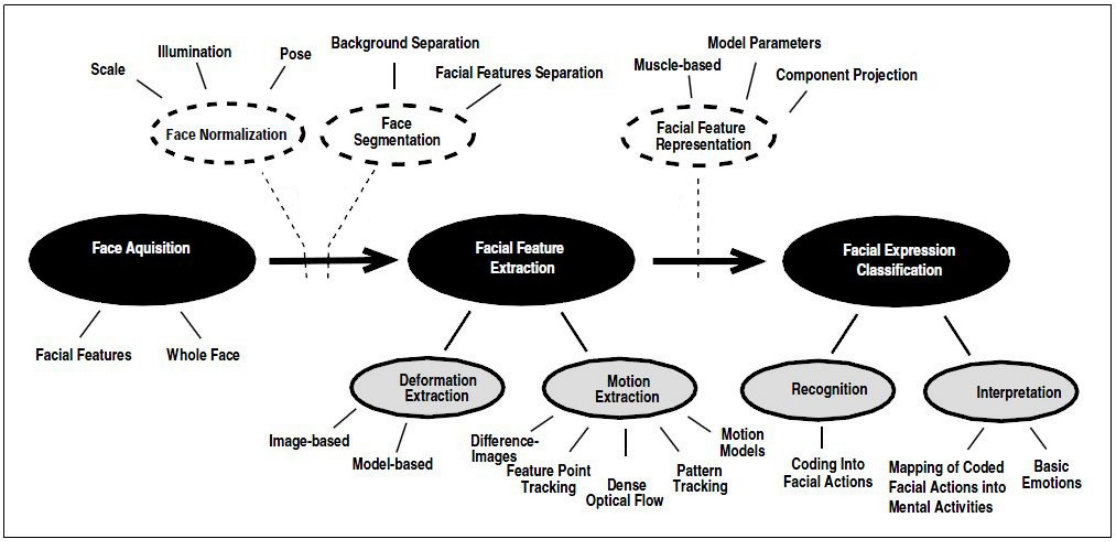


Figure 2.12: Generic facial expression analysis framework [36].

In recent years, research on using real-time tracking methods in facial expression recognition systems has become popular. First proposed by Edwards *et al.* [27], the AAM has attracted much interest in the computer vision community for modelling and segmenting deformable visual objects [87] [2].

Many researchers focus their work around the use of common standards, such as FACS and MPEG-4 (described in Section 2.4.2 and Section 2.4.3 respectively), to analyse facial expressions. A review of recent research work can be found in [61] and [36].

In this thesis we propose a novel representation for each universal facial expression as a collection of the most general quadratic transform coefficients that are model independent. Then we introduce a novel facial expression recognition technique based on a real-time facial feature tracking

[2] Active Appearance Models are explained in Appendix A.

system with the facial expression representations. In this context, the requirements are to use a facial feature tracking system that is marker-less and can generalise well to different faces. This has motivated us to use the AAM framework tools developed by Dr. Roland Goeke and Dr. Jason Saragih [89]. The framework supports three AAM fitting methods, in which some fitting methods are better to generalise than others. For example, the Simultaneous Inverse Compositional (SIC) method is known to generalise quite well and is hence often used when training a model that is to be used on other people's faces (or whether the user is simply unknown). The downside is that it is comparatively slow. On the other hand, for example, the Project Out Inverse Compositional (POIC) method is very good for person-specific models but not very good at all for person-independent. The discriminative-iterative (DI) approach, however, is almost as good as SIC in terms of generalisation, but also almost as fast as POIC. Therefore, in this thesis we use the AAM DI fitting method described in Appendix A.

2.6.2 Facial Caricatures

Facial caricature drawing exaggerates physical face features for a comical effect, and can create an entertaining, humorous and cartoon-like description of a person's face. Many skilled artists have contributed to the field of caricature illustrations and drawings, such as Hughes, Gair [48], while Redman [86] published work on how to draw caricatures and cartoons.

Recently, computerized caricatures have been introduced as part of computer graphics' non-photorealistic rendering technologies. Research has been conducted on how to produce caricatures automatically or semi-automatically

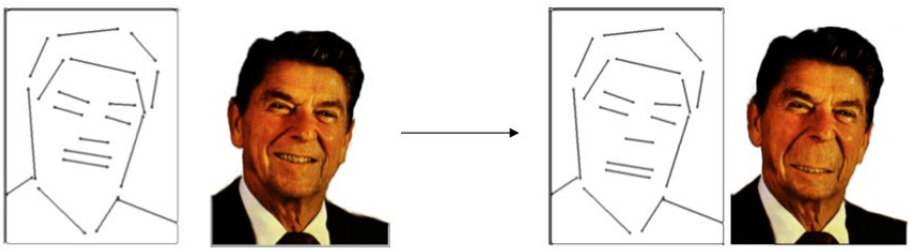


Figure 2.13: An example of a facial caricature generated using Akleman's method [2].

using various computer graphics techniques.

Generally, automatically or semi-automatically generated caricatures fall into two categories: (1) interactive and (2) example-based approaches. Interactive approaches rely on user input and include the work of Brenann [11] who presented the first interactive caricature generation method that produced exaggerated sketches. Fujiwara *et al.* [41] proposed, PICASSO, a template-based facial caricature system. Akleman [2] demonstrated a facial caricature system based on an interactive morphing tool. Using the interactive tool, the user can control several facial features. An example of Akleman's caricature method results is shown in Figure 2.13. Gooch *et al.* [43] presented a black-and-white illustration system based on the use of Gaussian filters and thresholding techniques. Their system is also capable of generating caricatures by imposing a deformable 2D grid over the surface of the face illustration. Users are then able to interactively change the shape of the whole facial illustration structure. Figure 2.14 shows an example of facial illustrations generated using their software. Most of the presented attempts achieve their non-photorealistic look by applying filtering algorithms or building tools for users to imitate traditional artistic styles.

Figure 2.14: An example of facial illustrations generated using the method proposed by Gooch *et al.* method [43].

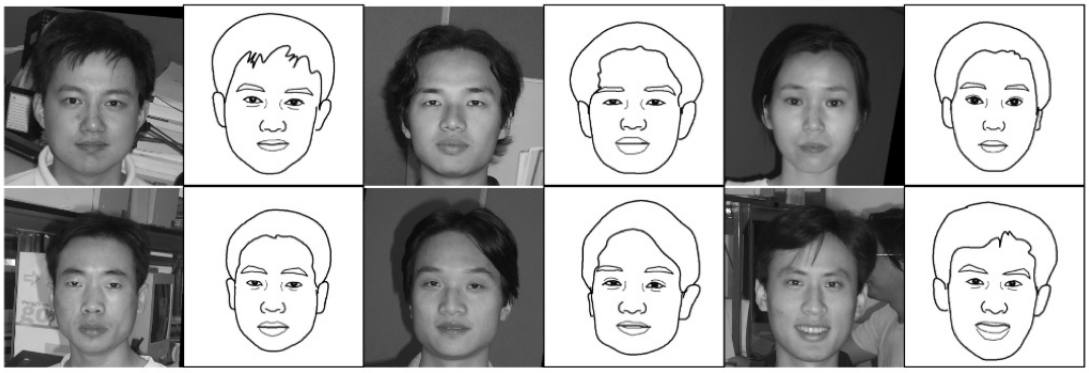


Figure 2.15: Caricature sketches generated by the example-based method proposed by Chen *et al.* [17].

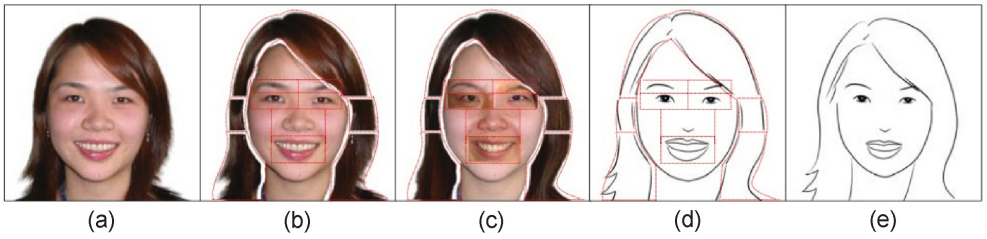


Figure 2.16: The process of the example-based caricature sketching by Chen *et al.* [16]. (a) Input image. (b) Subdivide the image into components. (c) Match the corresponding components found from the training example. (d) Corresponding sketch of the components in (c). (e) The final caricature drawings of the different components. The image and caption are obtained from [16].

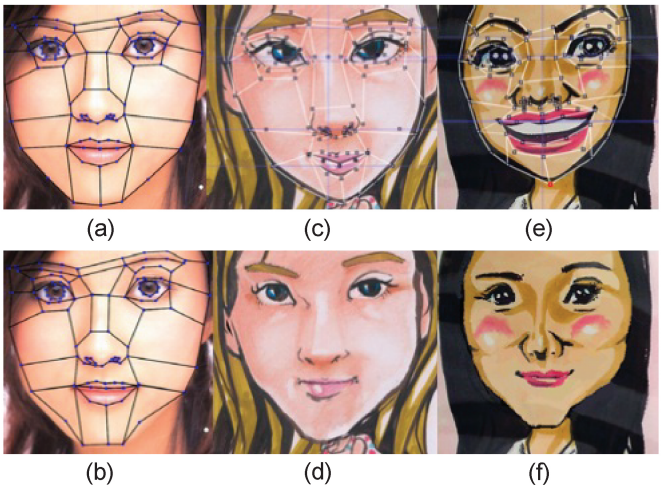


Figure 2.17: Examples of caricatures and exaggerations generated using Chiang *et al.* [19]. (a) Fitted mesh on the subject's image. (b) Mesh exaggeration. (c) An artist's sketch fitted with a mesh. (d) Caricature generated by exaggerating the mesh features in (c). (e) Another example of an artist's sketch. (f) Caricature of the art work in (e).

Example-based approaches include the work of Chen *et al.* [17] [18], who presented an example-based facial sketch generation system. Their work analyses and learns the sketch styles of artists and applies them to facial images. Figure 2.15 shows several caricature examples generated from their example-based method. Chen *et al.* [16] also proposed an example based method that uses two subsystems to handle the facial features and hair. Their idea is to decompose the face and hair features into components that are structurally related to each other [16]. Each of these components are then processed independently and recomposed to achieve the final result. Figure 2.16 shows the process proposed by Chen *et al.* [16]. Chiang *et al.* [19] showed a method for generating artistic caricatures used in defining facial feature points based on using active contour models and MPEG-4 FAPs. As input

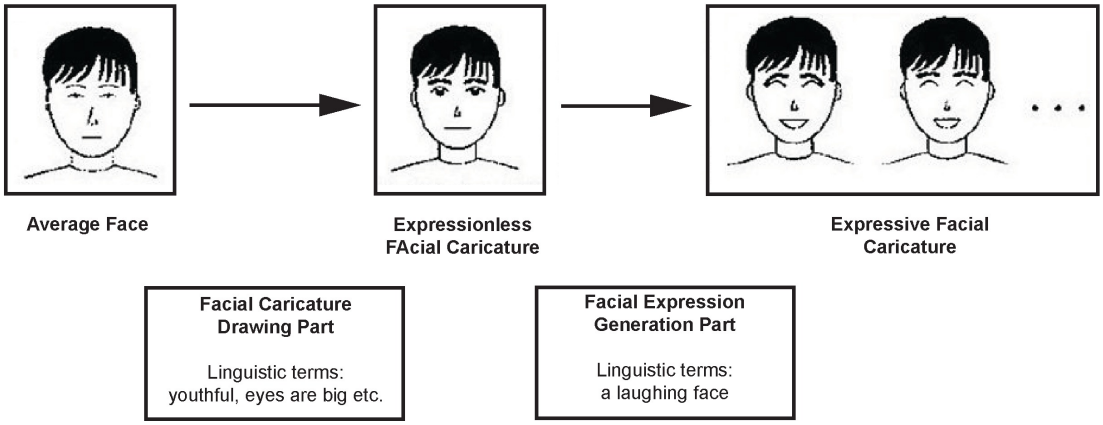


Figure 2.18: The process of the expressive caricature drawing proposed by Iwashita *et al.* [49].

they use an artist's finished work to locate and exaggerate the feature points. Figure 2.17 shows caricature examples generated by their system. Iwashita *et al.* [49] described a method for generating expressive facial caricatures based on facial expression linguistic terms. His work adds facial expressions to a neutral line caricature drawing of a face as shown in Figure 2.18. It is apparent that most example-based approaches have produced satisfactory results; however, one of the main limitations of example-based approaches is the large numbers of reference-caricature sketches drawn by artists that are needed to train these systems to create expressive caricatures.

In addition, several recent approaches have used statistical analysis and facial feature tracking techniques, such as active shape models, to extract the shape of the facial characteristics. For example, Mo *et al.* [66] produced a method to automatically locate facial features using AAM, and then to generate caricatures by feature normalisation and exaggeration. Figure 2.19 shows some caricature results generated by their system.

Figure 2.19: Caricatures generated from input images by the proposed method of Mo *et al.* [66].

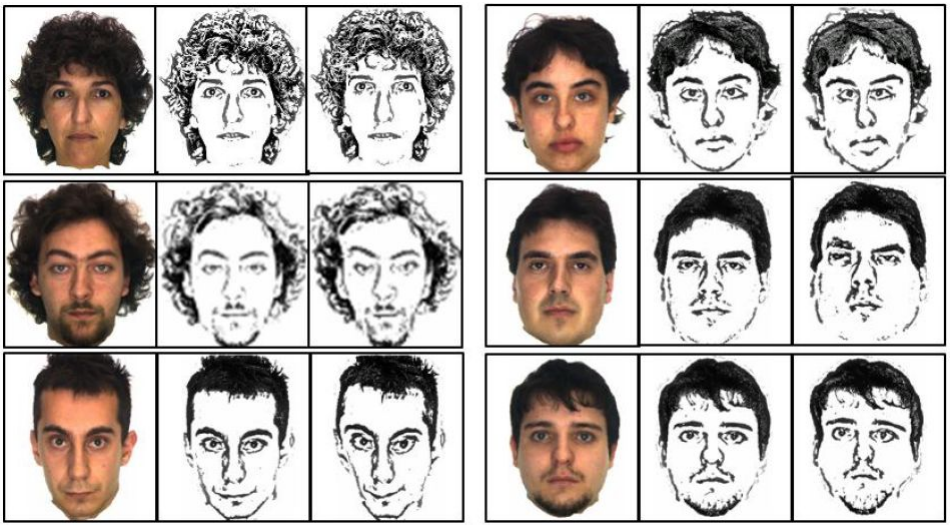


Figure 2.20: The results of the caricature generation system described by Tseng and Lien [101]. (Left) Input image. (Middle) Line-sketch. (Right) Exaggerated caricature.

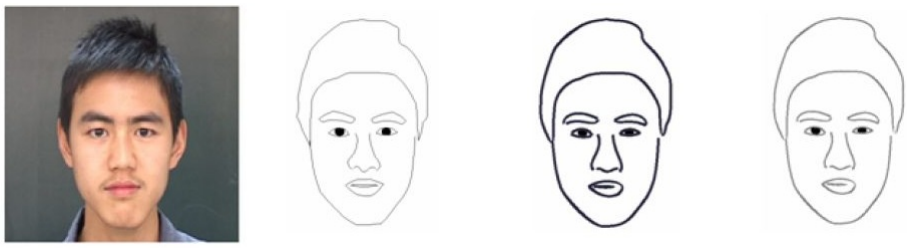


Figure 2.21: An example of the caricature sketch generated using the method proposed by Su *et al.* [96]. (Left) Input image. (Middle) Extracted drawing path. (Right) Two examples of caricatures with different line templates.

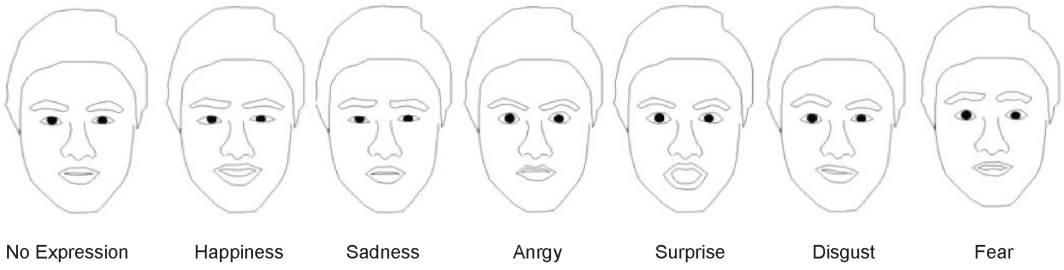


Figure 2.22: The sketched caricature results of the six main expressions presented by Su *et al.* [96].

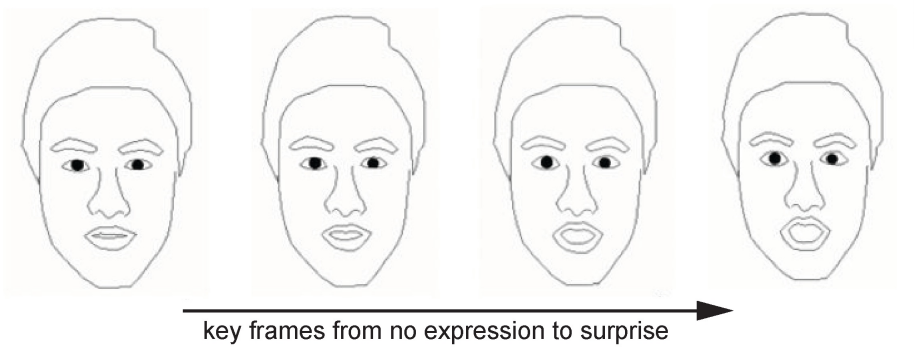


Figure 2.23: An animation sequence of the surprise expression generated using the method presented by Su *et al.* [96].

Tseng and Lien[101] developed a system to generate caricatures by statistically synthesizing facial features and applying a line-drawing sketch algorithm. Using the statistical synthesis of individual facial features, they were able to control the exaggeration of the features' shape and position (Figure 2.20 shows an example of caricatures generated using their method). Su *et al.* [96] produced a method to sketch faces and animate them. Their approach first generates a painting path using an active shape model and then applies free form deformation to a pen-and-ink line template for the path rendering. Figure 2.21 shows an example of the caricature drawing generated using their method, with two different line-templates. Su *et al.* also proposed a way to generate the six universal expressions from an input image as shown in Figure 2.22. Their system can also generate an animation sequence from neutral to any of the six expressions as shown in Figure 2.23. The work presented by Su *et al.* and a few other researchers [62] are the rare attempts that have been made to alter and control the facial expressions of the generated caricatures. In general, most approaches exaggerate the caricature's appearance by altering the overall facial shape based on capturing artists' prototypes.

In this thesis a novel expressive caricature generator method is proposed by using the AAM facial feature extraction system with novel mathematical model representation of facial expressions. The method takes a neutral facial image as an input and extracts its facial features. The extracted features are then deformed according to the deformation models to generate the expressiveness of the caricature appearance.

2.6.3 MPEG-4 Facial Animation

Expressive 3D virtual characters are an emerging technology metaphor amongst several sectors such as human computer interaction, games and e-learning. Introducing them in computerized systems simulates, as possible, the seamless natural interaction between the user and the computer.

Recently, the importance of emotional expression in 3D characters has been recognized and discussed. Expressive 3D characters can have several possible channels of expression such as gestures, facial expressions, gaze or posture [104]. Facial expressions are described to be the most expressive communicative channel [104][5].

In the past three decades, research and development in facial animation has advanced to include several standards that define how to perform facial expressions in synthetic 3D virtual characters. Many researchers have based their work on the FACS system [30]. However, animators generally find the FACS system to be an abstract standard, as it gives a verbal description of muscle movements.

As a successor to FACS, the MPEG-4 Facial Animation (FA) standard was designed to support the definition, encoding, transmission, and animation of 3D virtual models [72]. It plays an important role in many virtual character animation engines such as the Greta Embodied Conversational Agent (ECA) [78]. However, MPEG-4 FA is considered to be a low level parameterizations standard and an upper layer definition is required on the top of the MPEG-4 FA standard as described in Section 2.4.3.

Several approaches have been proposed to synthesize and animate facial expressions based on MPEG-4; this includes the work by Paradiso [73][74]

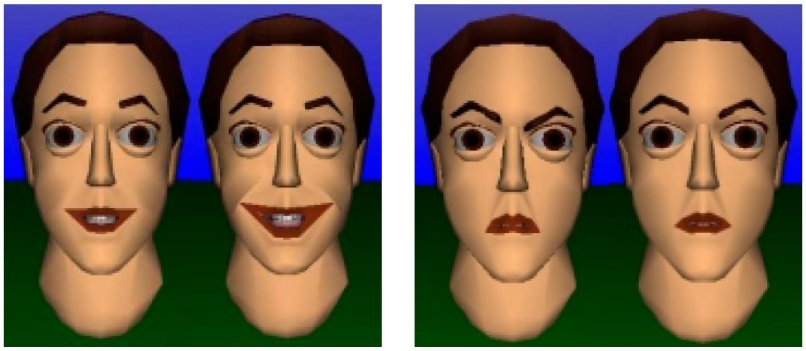


Figure 2.24: The facial expressions, a smile (left) and anger (right), generated with different intensities using algebraic operators of facial expressions proposed by Paradiso [74].

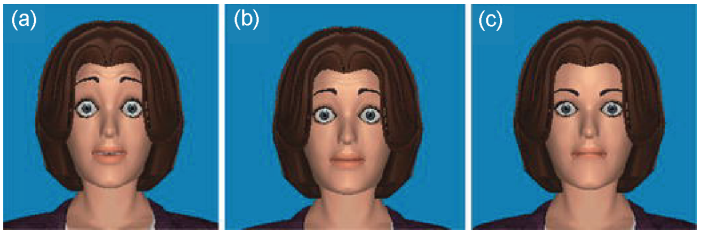


Figure 2.25: Intermediate expressions: (a) terrified, (b) afraid and (c) worried [64].

who proposed a method to define a set of algebra operators for combining facial animations and expressions based on MPEG-4 FAPs. Figure 2.24 shows an example of the smile and the angry expressions, mapped on an MPEG-4 compliant 3D face model. Malatesta *et al.* [64] used the rules of Raouzaiou *et al.* [84] to synthesis intermediate and primary expressions based on MPEG-4 FAPs. They used real measurements and took into account Whissel's study [109] to synthesize facial expressions on an ECA. Figure 2.25 illustrates some of the intermediate facial expressions results generated using the Greta ECA.

A challenge in synthesizing facial expressions is in producing a technique to generate and control facial animations on a 3D virtual character compliant with the MPEG-4 facial animation standard, while at the same time, the technique needs to be general enough to be used by other facial models. The lack of generalizing previously proposed methods is one of the main limitations, where most methods are dependent on the definition of the facial model.

In this thesis the synthesizing of facial expressions is described for MEPG-4 compliant facial models. The novelty of this work is being able to synthesize and control facial expressions using real muscle quadratic deformation parameters of facial expressions. The proposed approach allows us to synthesize not only the primary expression but also intermediate expressions.

Chapter III

Facial Expression Representation using Quadratic Deformation Models

In this chapter we present a novel approach to represent facial expressions in terms of mathematical transformation functions. The main advantage of our approach is the generic representation of facial expressions that can be employed in different applications areas such as facial animation and recognition.

The non-linear nature of muscle deformations can be represented using the rubber-sheet transformations described in Section 3.2. The muscle deformations for each facial expression are captured using the following steps:

1. Subdivide the face into 16 muscle based facial regions, as shown in Figure 3.2.

2. Using the most general rubber-sheet transformation of second degree (quadratic transformations), derive the deformation parameters for each region by applying the least squares fitting technique.

3. Construct Facial Deformation Tables (FDT) to mathematically represent facial expressions.

3.1 Defining Facial Regions

To define facial regions that can be used to represent facial expressions mathematically, the FACS description of facial expressions [30] and the anatomy description of facial muscles [35] [38] were studied.

Faigan [35] stated in his book that facial expressions are mainly driven by eleven muscles out of the twenty six main facial muscles. These muscles are [35] [50] (1) Orbicularis oculi, (2) Levator palpebrae, (3) Levator labii superioris, (4) Zygomatic major, (5) Risorius/Platysma, (6) Frontalis, (7) Orbicularis oris, (8) Corrugator, (9) Depressor anguli oris, (10) Depressor labii inferioris, and (11) Mentalis. Figure 3.1 shows an illustrations of the eleven muscles.

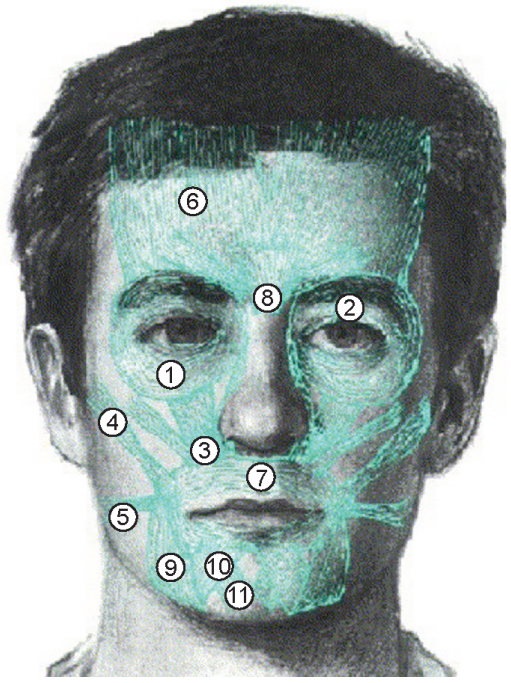


Figure 3.1: The eleven muscles responsible for facial expressions that are presented by Faigan [35]. Image adopted with permission from the author [35].

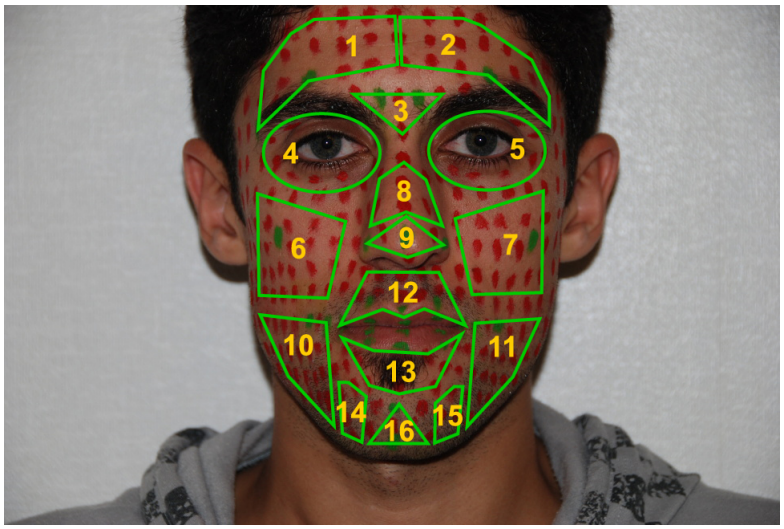


Figure 3.2: The sixteen muscle based facial regions.

Using the above resources, we defined sixteen regions that represent the deformation of the facial expressions. The sixteen regions form the minimum number for defining independent facial muscle groups. Figure 3.2 illustrates the defined facial regions.

3.2 Transformations

3.2.1 Affine Transformations

Affine transformations or first-order linear transformations are defined in [108] as "any transformation that preserves colinearity (i.e., all points lying on a line initially still lie on a line after transformation) and ratios of distances (e.g., the midpoint of a line segment remains the midpoint after transformation)". Affine transformations are represented by six parameters that allow for the translation, rotation, scaling, and shearing operations. Figure 3.3 shows an example of several affine transformations and the equations

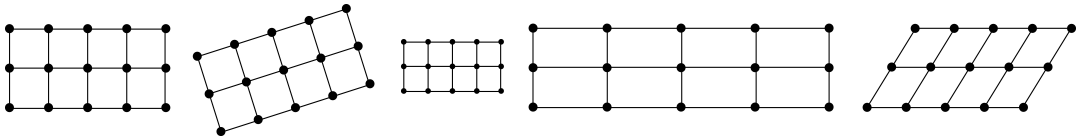


Figure 3.3: Affine transformations.

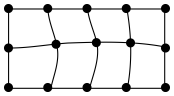


Figure 3.4: High-order polynomial transformations.

that represent an affine transformation in two dimensions are as follows:

$$x'_i = A_1 x_i + A_2 y_i + A_3 \quad (3.1)$$

$$y'_i = B_1 x_i + B_2 y_i + B_3 \quad (3.2)$$

where,

$i : 1, ..., n$

n : number of transformed points

$(A_j, B_j), j = 1, .., 3$: the transformation parameters.

3.2.2 Rubber-sheet Transformations (High-order Polynomial Transformations)

Rubber-sheet transformations are higher-order (non-linear) polynomial transformations [107, 42]. The name comes from the analogy of stretching an elastic piece of rubber to fit over a surface shape. Unlike affine transformations, high-order transformations can result in shape nonlinearity as shown in Figure 3.4.

A simple plane transformation is the mapping of a geometric point (x, y) in a coordinate plane to another position (x'_i, y'_i). Figure 3.5 shows an example of such a transformation. Polynomial transformations are one form of plane transformations and they are computationally simple to approximate. In the two-dimensional space, rubber-sheet transformations are defined by the second-order polynomial equations:

$$x'_i = a_1 x_i^2 + a_2 x_i y_i + a_3 y_i^2 + a_4 x_i + a_5 y_i + a_6 \quad (3.3)$$

$$y'_i = b_1 x_i^2 + b_2 x_i y_i + b_3 y_i^2 + b_4 x_i + b_5 y_i + b_6 \quad (3.4)$$

where,

$i : 1, ..., n$

n : number of transformed points

$(a_j, b_j), j = 1, ..., 6$: the transformation parameters.

The transformed point (x'_i, y'_i) is computed by substituting the (x, y) point values into the right side of the equations.

Geographical Information Systems (GIS) use rubber-sheet transforma-

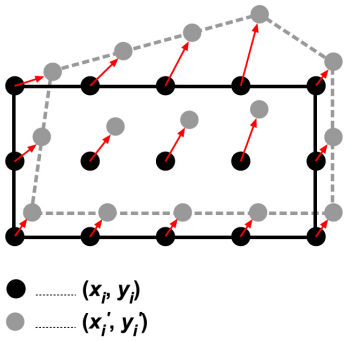


Figure 3.5: 2D points transformations.

tions for operations such as converting map coordinates, map alignments and geometrical correction between maps [91].

The fact that the surface of the face (facial skin) is an elastic deformable layer allows us to use rubber-sheet transformation to represent facial expressions. One of the initial studies to show that using the facial skin movements and motion for facial expression analysis was the work by Bassili [8].

3.3 Rubber-sheet Transformations to Derive the Facial Expression Deformation Parameters

To use second-order rubber-sheet equations (quadratic transformations) for defining the expression model, it is assumed that the 12 transformation parameters $((a_j, b_j), j = 1, ..., 6)$ are not known; however, if the coordinate points before and after the transformation are known (i.e. (x'_i, y'_i) and (x_i, y_i)) as in Figure 3.5, then with enough coordinate points, the *solution* for the 12 parameters that best fit the transformation can be worked out. This is done by solving from the known coordinate points using the least-square approach [107]. The process of finding the solution is described in Figure 3.6.

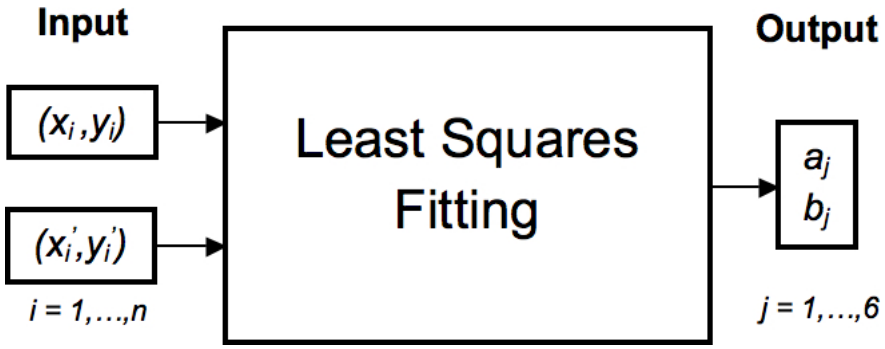


Figure 3.6: Derivation of the deformation parameters.

To find the 12 parameters representing a transformation, the minimum of the sum of squares (or the least squares fitting) [107] is carried out on the pair of the quadratic Equations 3.3 and 3.4, which represent the x and y rubber-sheet transformations.

To minimize the errors between the transformed coordinates and the measured coordinates of points for a given expression, the minimum of the sum of error-squares is found. The minimum of the sum of error-squares is achieved when the gradient of the sum-of-error-squares for the quadratic transformation equations is zero. The following equations represent the sum-of-error-squares:

$$G_x = \sum \left[x'_i - (a_1 x_i^2 + a_2 x_i y_i + a_3 y_i^2 + a_4 x_i + a_5 y_i + a_6)\right]^2 \quad (3.5)$$

$$G_y = \sum \left[y'_i - (b_1 x_i^2 + b_2 x_i y_i + b_3 y_i^2 + b_4 x_i + b_5 y_i + b_6)\right]^2 \quad (3.6)$$

The minimum of the sum of squares is found by setting the gradient of G_x and G_y to zero with respect to all unknown variables as follows:

$$\frac{\partial G_x}{\partial a_j} = 0; \frac{\partial G_y}{\partial b_j} = 0 \qquad j = 1, ...6. \quad (3.7)$$

Differentiating G_x and G_y with respect to all unknown variables and equating it to zero gives the conditions for the minimum error in the transform coefficients in the least-square sense. For example, differentiating G with respect to $a_1 (\frac{\partial G_x}{\partial a_1} = 0)$ is carried out as follows:

$$
\begin{aligned}
2\sum_{i=1}^{n}\Big[x_i' - (a_1x_i^2 + a_2x_iy_i + a_3y_i^2 + a_4x_i + a_5y_i + a_6)\Big]x_i^2 = 0 \\
\sum_{i=1}^{n}\Big[x_i'x_i^2 - a_1x_i^4 - a_2x_i^3y_i - a_3x_i^2y_i^2 - a_4x_i^3 - a_5x_i^2y_i - a_6x_i^2)\Big] = 0 \\
\sum_{i=1}^{n}x_i'x_i^2 - \sum_{i=1}^{n}a_1x_i^4 - \sum_{i=1}^{n}a_2x_i^3y_i - \sum_{i=1}^{n}a_3x_i^2y_i^2 - \sum_{i=1}^{n}a_4x_i^3 - \sum_{i=1}^{n}a_5x_i^2y_i - \sum_{i=1}^{n}a_6x_i^2 = 0
\end{aligned}
\tag{3.8}
$$

This can be written as:

$$
n_{1x} - a_1m_{11} - a_2m_{12} - a_3m_{13} - a_4m_{14} - a_5m_{15} - a_6m_{16} = 0 \tag{3.9}
$$

where each element in the derivate is computed as shown in Table 3.1. The table also implies to the derivative elements of G_y, where the computed values for $n_{1y}, ..., n_{6y}$ are also given.

For the derivative of G_x with respect to the other unknown variables, the same process is applied to obtain the values for n_ix and m_{ij}, where $i = 1, ...6$; $j = 1, ...6$. Having found those values we can now solve for the parameter values $P_a(a_1, ..., a_6)$ and $P_b(b_1, ..., b_6)$ by solving the following matrix setup $PM = N, P = M^{-1}N$

Table 3.1: Matrix elements used to solve for the unknown variables of the quadratic Equation 3.3 and Equation 3.4.

$n_{1x} = \sum_{i=1}^{n} x'_i x_i^2$	$n_{2x} = \sum_{i=1}^{n} x'_i x_i y_i$	$n_{3x} = \sum_{i=1}^{n} x'_i y_i^2$	$n_{4x} = \sum_{i=1}^{n} x'_i x_i$	$n_{5x} = \sum_{i=1}^{n} x'_i y_i$	$n_{6x} = \sum_{i=1}^{n} x'_i$
$n_{1y} = \sum_{i=1}^{n} y'_i x_i^2$	$n_{2y} = \sum_{i=1}^{n} y'_i x_i y_i$	$n_{3y} = \sum_{i=1}^{n} y'_i y_i^2$	$n_{4y} = \sum_{i=1}^{n} y'_i x_i$	$n_{5y} = \sum_{i=1}^{n} y'_i y_i$	$n_{6y} = \sum_{i=1}^{n} x'_i$
$m_{11} = \sum_{i=1}^{n} x_i^4$	$m_{12} = \sum_{i=1}^{n} x_i^3 y_i$	$m_{13} = \sum_{i=1}^{n} x_i^2 y_i^2$	$m_{14} = \sum_{i=1}^{n} x_i^3$	$m_{15} = \sum_{i=1}^{n} x_i^2 y_i$	$m_{16} = \sum_{i=1}^{n} x_i^2$
$m_{21} = \sum_{i=1}^{n} x_i^3 y_i$	$m_{22} = \sum_{i=1}^{n} x_i^2 y_i^2$	$m_{23} = \sum_{i=1}^{n} x_i y_i$	$m_{24} = \sum_{i=1}^{n} x_i^2 y_i$	$m_{25} = \sum_{i=1}^{n} x_i y_i^2$	$m_{26} = \sum_{i=1}^{n} x_i y_i$
$m_{31} = \sum_{i=1}^{n} x_i^2 y_i^2$	$m_{32} = \sum_{i=1}^{n} x_i y_i^3$	$m_{33} = \sum_{i=1}^{n} y_i^4$	$m_{34} = \sum_{i=1}^{n} x_i y_i^2$	$m_{35} = \sum_{i=1}^{n} y_i^3$	$m_{36} = \sum_{i=1}^{n} y_i^2$
$m_{41} = \sum_{i=1}^{n} x_i^3$	$m_{42} = \sum_{i=1}^{n} x_i^2 y_i$	$m_{43} = \sum_{i=1}^{n} x_i y_i^2$	$m_{44} = \sum_{i=1}^{n} x_i^2$	$m_{45} = \sum_{i=1}^{n} x_i y_i$	$m_{46} = \sum_{i=1}^{n} x_i$
$m_{51} = \sum_{i=1}^{n} x_i^2 y_i$	$m_{52} = \sum_{i=1}^{n} x_i y_i^2$	$m_{53} = \sum_{i=1}^{n} y_i^3$	$m_{54} = \sum_{i=1}^{n} x_i y_i$	$m_{55} = \sum_{i=1}^{n} y_i^2$	$m_{56} = \sum_{i=1}^{n} y_i$
$m_{61} = \sum_{i=1}^{n} x_i^2$	$m_{62} = \sum_{i=1}^{n} x_i y_i$	$m_{63} = \sum_{i=1}^{n} y_i^2$	$m_{64} = \sum_{i=1}^{n} x_i$	$m_{65} = \sum_{i=1}^{n} y_i$	$m_{66} = 1$

To illustrate, the matrix elements to solve for the parameter values $P_a(a_1, ..., a_6)$ are as follows

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} & m_{15} & m_{16} \\ m_{21} & m_{22} & m_{23} & m_{24} & m_{25} & m_{26} \\ m_{31} & m_{32} & m_{33} & m_{34} & m_{35} & m_{36} \\ m_{41} & m_{42} & m_{43} & m_{44} & m_{45} & m_{46} \\ m_{51} & m_{52} & m_{53} & m_{54} & m_{55} & m_{56} \\ m_{61} & m_{62} & m_{63} & m_{64} & m_{65} & m_{66} \end{bmatrix}^{-1} \begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \\ n_5 \\ n_6 \end{bmatrix}$$

The solution to the set of parameters can now be used to obtain the transformed coordinate (x'_i, y'_i) of any point by substituting the (x_i, y_i) coordinates into Equation 3.3 and Equation 3.4. In order to derive a solution it is necessary to have at least six points in each muscle region.

3.4 Method

3.4.1 Mathematical Representations of Facial Expressions

Mathematically, the questions we are trying to answer are: How does a region deform for a particular expression (Figure 3.7), and how can we model the deformation properties? To answer these questions, the defined facial regions are represented as non-linear shapes that can deform into any other shape depending on the facial expression. To find out how each of regions deform, several face markers are placed on an actor's face and used to track how each region is deformed from the neutral facial expression to any performed expression (surprise, fear, anger, sadness, disgust, and smile). Figure 3.8 illustrates an actor performing a facial expression. To find the region

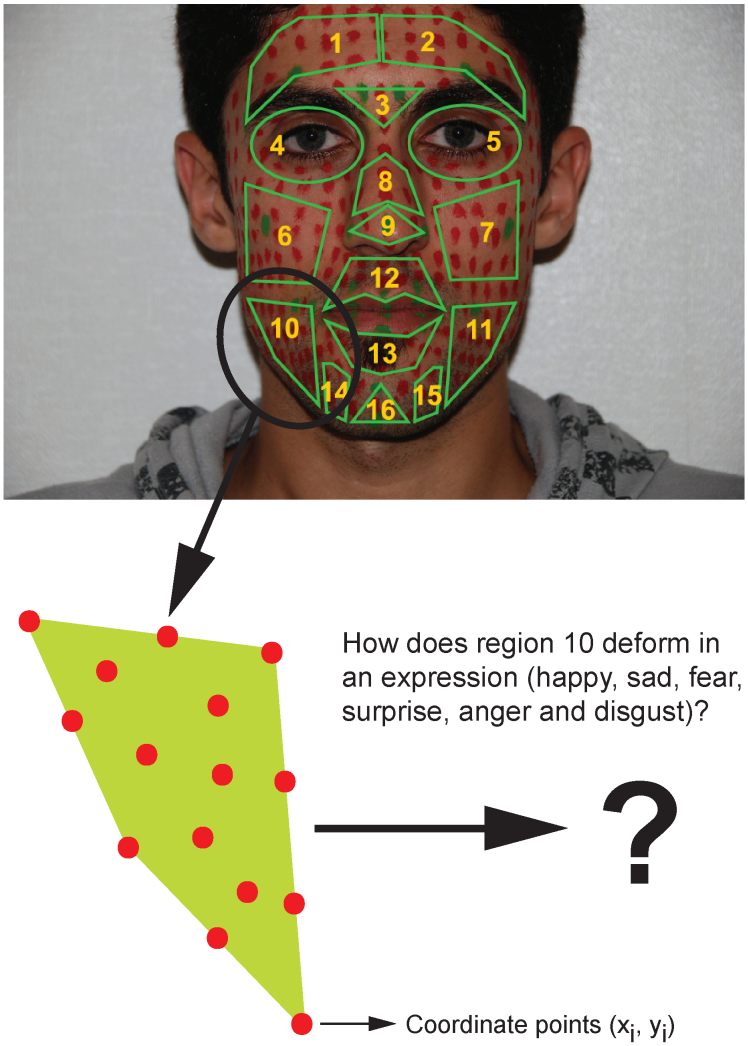


Figure 3.7: An example to demonstrate region deformations.

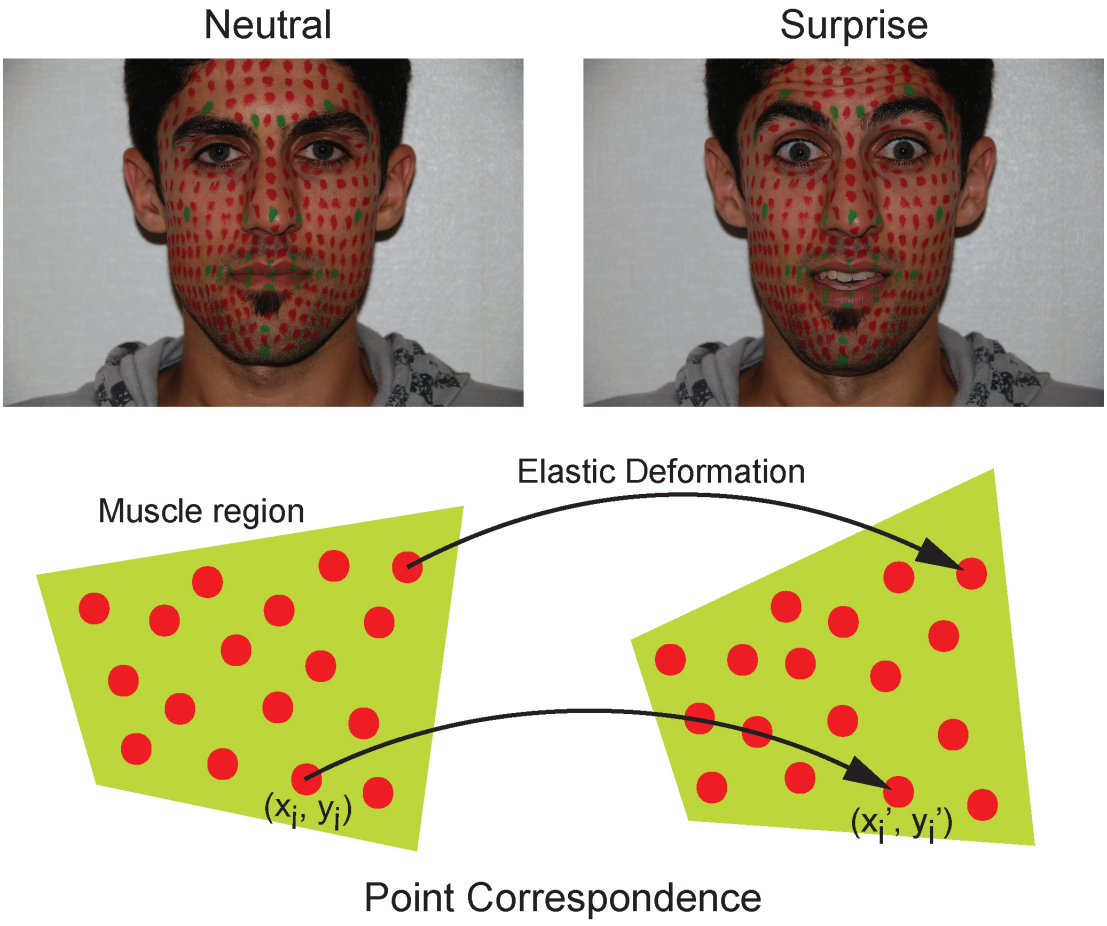


Figure 3.8: An example of an actor performing the surprise expression.

deformation parameters using the rubber-transformations (described in Section 3.3), we collect, from 2D images, the (x, y) coordinates for the neutral expression and then collect the corresponding coordinates (x', y') for the preformed expression (i.e. $x_i \Rightarrow x_i^{'}$; $y_i \Rightarrow y_i^{'}$ $i = 1, ..., n$ where n represent the number of points in a region). The data is normalized to eliminate all global head movements by pose- and scale-normalisations, as described in Section 3.4.2. Using the normalized data, we compute the parameter values for each facial region to obtain the following set of parameters for an expression
$(a_1, ..., a_6, b_1, ..., b_6)_k$ $(k = 1, ..., 16)$
where k is the facial region's number. Using the computed deformation parameters for region k, we can obtain the transformed coordinates $(x_i^{'}, y_i^{'})_k$ of any point within a region by substituting the parameter values
$(a_1, ..., f_6, b_1, ..., b_6)_k$ $(k = 1, ..., 16)$ and the original coordinate points $(x_i, y_i)_k$ into the right hand-side of the quadratic Equations 3.3 and 3.4.

Therefore, applying this process to all facial regions will result in deforming the facial regions from the neutral facial expression to the desired facial expression.

3.4.2 Normalisation Process

To analyze the details of the facial regions, global head movements are eliminated by pose- and scale- normalisations. The normalisation is achieved based on two global facial locations (eye pupils) p_1 and p_2 as shown in Figure 3.9, where m is the middle point between p_1 and p_2.

The line (m, n) is perpendicular to the line (p_1 and p_2) and has the same distance as the line (m, p_1). All of the points are normalized through the

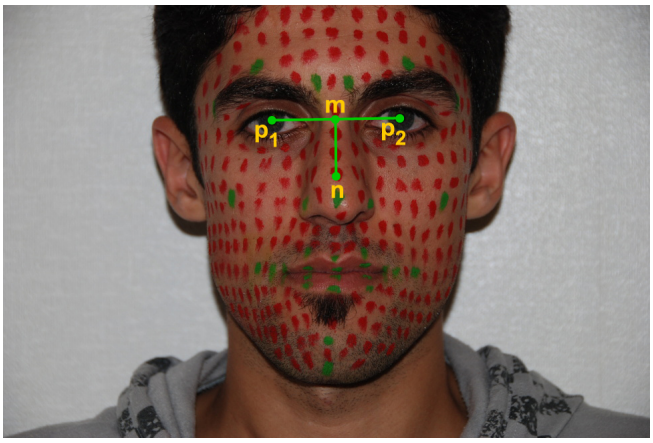


Figure 3.9: Global facial location points used in the normalisation process.

following three transforms:

- Translate point m to the origin, and then translate all the other points about the origin by $(-m_x, -m_y, -m_z)$ as illustrated in Figure 3.10 (i).

- Rotate all points about the origin by θ around the Z-axis, where θ is the angle between vector (p_1, p_2) and the X-axis (see Figure 3.10 (ii)).

- Scale all the points to $(\frac{x_i}{x_{p_2}}, \frac{y_i}{x_{p_2}})$ as shown in Figure 3.10 (iii).

This operation will scale points p_1 and p_2 to $(-1, 0)$ and $(1, 0)$ respectively, and point n to $(0, -1)$.

3.4.3 Facial Deformation Tables (FDTs)

The aim of our work is to develop a new method for representing facial expressions in terms of deformation parameters, where those parameters can be employed in facial applications areas such as facial animations, recognition and interpretations. To achieve the facial expression representations,

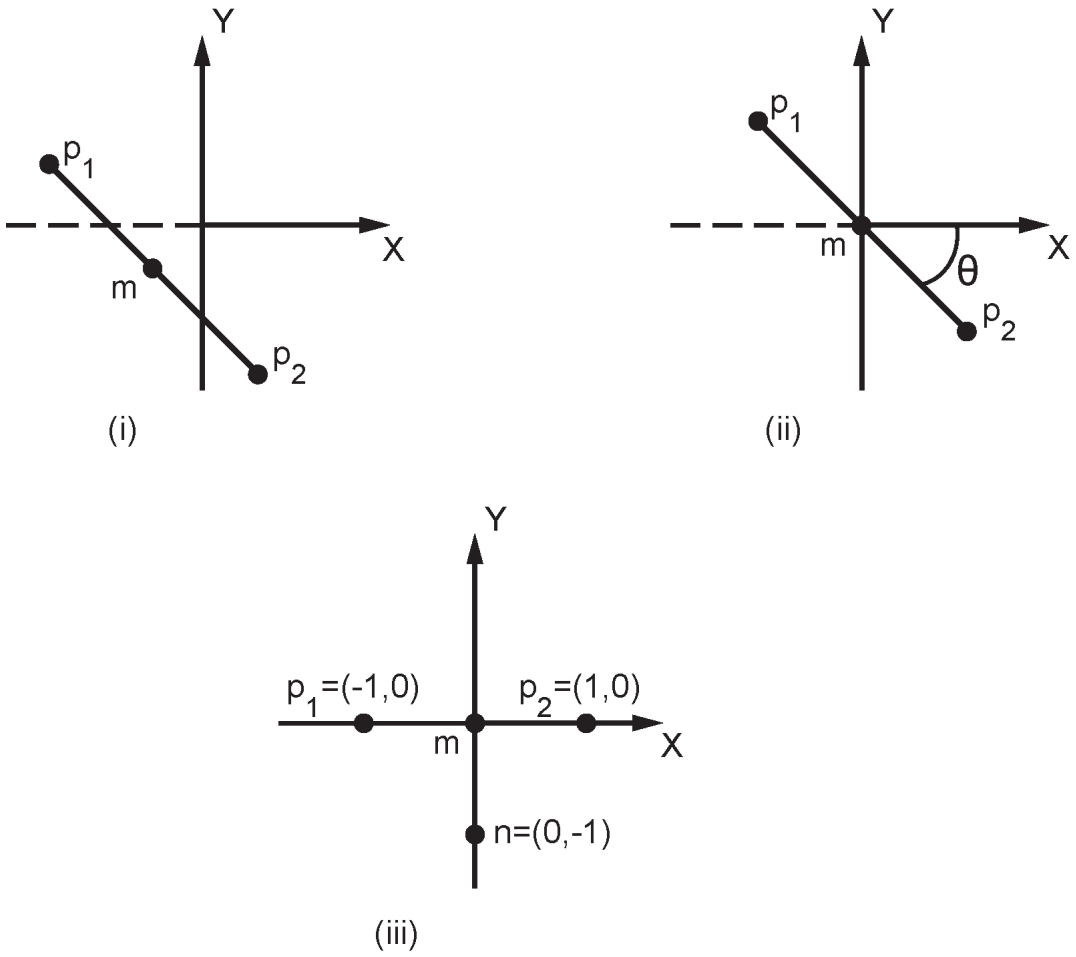


Figure 3.10: Normalisation operations: (i) translating the points about the origin, (ii) eliminating the XY rotations and (iii) scaling operation.

we defined a *Facial Deformation Table* (FDT_E), for an expression E, that represents the deformation parameters for each facial region of expression E. The following describes the procedure to compute a FDT_E for each of the six main expressions (surprise, fear, anger, sadness, disgust, and smile).

- **Facial expression data acquisition:** The facial expressions of twelve model subjects, 20-50 years old, were analyzed. Each subject was asked to perform the six main facial expressions from the neutral facial expression state. A series of frontal photographs of the face were taken to capture each of the expressions. The subjects had several markers positioned on their faces for the duration of the photography.

- **Collecting Coordinate points:** The image data was analyzed for each facial expression by collecting the coordinate points for the neutral expression and the performed expression (before and after the expression). The collected coordinate points were grouped into their facial regions as described in Section 3.1.

- **Normalisation:** All of the global head movements are eliminated by pose and scale normalisations. The normalisation is achieved based on two global facial locations (eye pupils). This process is described further in Section 3.4.2.

- **Computing the deformation parameters:** For each of the facial expression coordinate data sets, the deformation parameter values were computed as described in Section 3.3. This computed result forms the FDT_E for expression E.

3.5 Results

The results from the procedure described in Section 3.4.3 gives a set of FDT_E that represents the deformation parameters for each of the participants' facial expressions. To compute a generic set of FDT_E that can be used to represent expressions, the mean μ of the FDT_E parameter values for expression E are computed. The computed results represent each of the six main expressions in a FDT^{μ}_{E}. Tables 3.2-3.7 show the FDT^{μ}_{smile}, FDT^{μ}_{fear}, FDT^{μ}_{anger}, $FDT^{\mu}_{surprise}$, FDT^{μ}_{sad} and $FDT^{\mu}_{disgust}$ respectively. From the FDT tables, it is observed that the values of the coefficients a_6 and b_6 of the quadratic equations are always near zero. This implies that facial expressions can be adequately represented using 10 coefficient parameters (a_1-a_5 and b_1-b_5) and the translation coefficients (a_6 and b_6) can be neglected. Figure 3.11 shows the results of using the FDT parameters to transform facial region points.

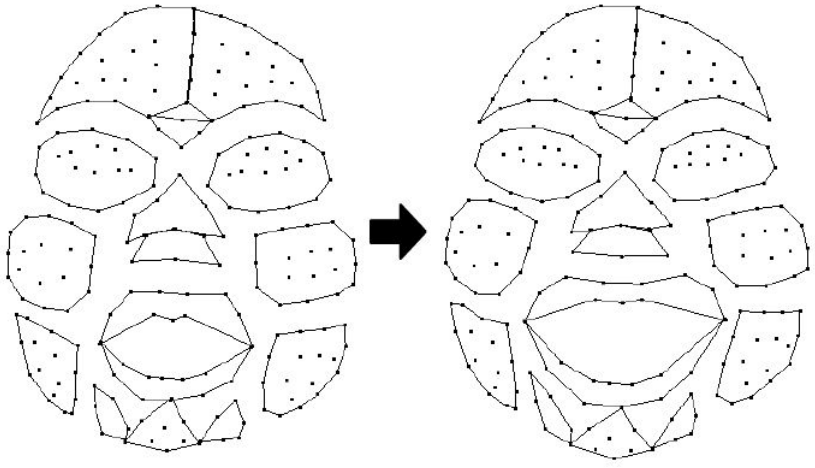


Figure 3.11: An example of facial region points deforming to a 'smile' facial expression. The points before and after the deformation are used to derive the deformation parameters.

Table 3.2: Smile facial deformation table.

Region	a_1	a_2	a_3	a_4	a_5	a_6	b_1	b_2	b_3	b_4	b_5	b_6
1	-0.003	0.009	-0.002	0.991	0.007	0.000	-0.007	0.015	0.011	-0.018	0.987	0.000
2	-0.002	0.000	-0.007	1.005	0.015	0.000	0.005	0.005	0.004	-0.021	0.998	0.000
3	-0.016	-0.119	0.093	1.070	-0.045	-0.001	0.009	-0.037	0.073	-0.009	0.951	0.001
4	-0.009	-0.019	-0.104	0.977	-0.024	0.000	-0.009	0.086	0.101	-0.015	0.994	0.001
5	0.001	-0.005	0.104	1.001	0.026	-0.001	-0.012	-0.126	0.089	0.021	1.037	0.001
6	0.040	-0.041	-0.026	1.054	-0.012	0.000	-0.078	0.022	0.003	-0.194	1.012	0.000
7	-0.050	-0.088	-0.031	1.035	-0.048	0.000	-0.082	-0.030	-0.015	0.197	0.992	0.000
8	0.064	-0.035	-0.022	0.978	-0.023	-0.001	0.132	-0.021	0.057	-0.022	1.072	0.004
9	-0.024	-0.207	0.010	0.803	0.008	0.000	0.247	-0.059	0.010	-0.088	1.048	0.000
10	0.157	-0.007	0.022	1.387	-0.012	0.000	-0.113	-0.028	-0.020	-0.286	0.931	0.000
11	-0.107	-0.008	-0.048	1.222	-0.112	0.000	-0.132	0.022	-0.028	0.314	0.911	0.000
12	0.013	0.050	-0.006	1.308	-0.012	0.000	0.107	0.018	0.007	0.047	0.961	0.000
13	0.021	0.065	0.005	1.326	0.015	0.000	0.306	0.057	-0.023	0.133	0.972	0.000
14	0.318	0.061	0.003	1.609	-0.014	0.000	-0.173	-0.322	0.073	-1.521	1.315	0.000
15	-0.321	0.066	0.003	1.673	0.042	0.000	-0.218	0.146	0.038	1.052	1.222	0.000
16	0.044	0.153	0.005	1.632	0.015	0.000	0.003	0.036	0.023	0.042	1.110	0.000

Table 3.3: Fear facial deformation table.

Region	a_1	a_2	a_3	a_4	a_5	a_6	b_1	b_2	b_3	b_4	b_5	b_6
1	-0.025	0.019	-0.013	0.921	0.006	0.000	-0.031	0.046	-0.068	-0.076	1.122	0.000
2	0.027	0.019	0.025	0.927	-0.036	0.000	-0.008	-0.041	-0.088	0.030	1.158	0.000
3	0.003	0.347	0.007	0.763	-0.032	0.000	0.142	0.007	-0.282	-0.064	1.283	0.000
4	0.018	-0.024	-0.028	1.019	-0.011	0.000	-0.059	-0.001	-0.078	-0.120	1.028	-0.002
5	0.006	-0.008	-0.006	0.980	-0.004	0.000	-0.063	0.021	-0.042	0.133	1.033	-0.001
6	0.015	-0.006	0.023	1.016	0.022	0.000	0.021	-0.015	-0.028	0.056	0.912	0.000
7	0.037	0.033	-0.024	0.929	-0.082	0.000	0.032	0.014	-0.041	-0.085	0.883	0.000
8	-0.021	-0.015	0.007	0.966	0.022	0.003	0.077	0.001	-0.093	0.013	0.867	-0.010
9	-0.026	-0.028	0.021	0.958	0.033	0.000	-0.056	0.168	-0.015	0.206	0.944	0.000
10	-0.020	0.026	-0.008	0.975	-0.001	0.000	-0.008	0.049	-0.018	0.065	0.991	0.000
11	0.056	0.028	0.007	0.881	-0.030	0.000	0.040	-0.027	-0.018	-0.132	0.959	0.000
12	0.003	0.050	0.003	1.092	0.011	0.000	-0.084	-0.021	-0.024	-0.040	0.934	0.000
13	0.015	-0.100	0.003	0.728	0.012	0.000	0.225	-0.010	0.059	-0.040	1.242	0.000
14	-0.142	-0.008	0.005	0.699	0.060	0.000	0.150	-0.213	0.060	-0.518	1.213	0.000
15	0.226	-0.081	0.002	0.510	0.039	0.000	-0.034	-0.048	0.029	-0.011	1.170	0.000
16	-0.013	0.062	-0.025	1.143	-0.120	0.000	0.097	-0.036	0.005	-0.124	1.048	0.000

Table 3.4: Anger facial deformation table.

Region	a_1	a_2	a_3	a_4	a_5	a_6	b_1	b_2	b_3	b_4	b_5	b_6
1	-0.104	0.037	-0.052	0.751	0.058	0.000	0.090	-0.009	0.029	0.166	0.950	0.000
2	0.090	0.023	0.045	0.795	-0.051	0.000	0.088	0.017	0.038	-0.158	0.929	0.000
3	0.015	-1.003	-0.344	1.259	0.181	0.000	-0.184	-0.337	-0.021	0.157	0.926	0.000
4	0.024	-0.079	0.254	1.025	0.033	0.000	-0.031	0.040	-0.366	-0.043	0.820	0.001
5	-0.010	0.033	-0.288	1.008	-0.051	0.000	-0.040	0.006	-0.278	0.049	0.771	0.000
6	-0.013	-0.010	0.003	0.973	0.010	0.000	-0.001	-0.007	-0.003	-0.014	0.988	0.000
7	0.023	0.014	0.013	0.965	-0.004	0.000	-0.007	0.000	0.003	0.030	1.018	0.000
8	0.072	-0.015	0.023	0.992	0.043	0.000	0.082	0.013	-0.034	0.005	0.978	-0.002
9	-0.040	-0.046	0.048	0.959	0.056	0.000	0.090	0.065	-0.014	0.060	0.995	0.000
10	0.016	-0.025	0.007	1.012	-0.002	0.000	0.045	0.005	-0.017	0.137	0.924	0.000
11	-0.005	-0.006	-0.003	1.023	0.008	0.000	0.052	0.015	-0.004	-0.101	0.958	0.000
12	0.005	0.035	-0.009	1.088	-0.011	0.000	0.014	-0.009	0.009	-0.022	1.030	0.000
13	-0.004	-0.026	-0.011	0.938	-0.025	0.000	-0.067	0.022	-0.016	0.032	0.939	0.000
14	0.162	0.055	-0.024	1.365	-0.084	0.000	0.299	-0.086	-0.003	0.236	0.925	0.000
15	-0.080	0.020	0.018	1.220	0.085	0.000	0.006	0.120	0.038	0.331	1.109	0.000
16	0.060	0.059	-0.002	1.189	-0.001	0.000	0.010	0.047	0.003	0.146	1.008	0.000

Table 3.5: Surprise facial deformation table.

Region	a_1	a_2	a_3	a_4	a_5	a_6	b_1	b_2	b_3	b_4	b_5	b_6
1	0.030	0.009	0.003	1.064	0.002	-0.001	-0.109	0.053	-0.162	-0.225	1.296	0.000
2	-0.007	0.027	-0.017	1.017	0.025	0.000	-0.097	-0.063	-0.167	0.197	1.304	-0.001
3	0.012	0.974	0.330	0.579	-0.144	0.000	0.148	0.018	-1.396	0.001	2.032	-0.002
4	0.005	-0.062	-0.131	1.011	-0.142	0.000	-0.058	-0.102	0.345	-0.147	1.210	-0.003
5	-0.009	-0.020	0.117	1.026	0.105	0.000	-0.062	0.103	0.287	0.150	1.179	-0.004
6	0.005	-0.002	0.016	1.019	0.009	0.000	0.029	-0.001	-0.039	0.083	0.912	0.000
7	0.011	0.002	-0.028	0.988	-0.043	0.000	0.010	-0.005	-0.042	-0.045	0.933	0.000
8	0.009	-0.068	-0.033	0.947	-0.028	-0.002	0.064	-0.119	-0.116	-0.061	0.848	-0.014
9	0.008	0.000	-0.009	0.990	-0.011	0.000	-0.080	-0.014	-0.034	-0.098	0.921	0.000
10	-0.049	0.057	-0.019	0.974	0.000	0.000	0.018	0.026	-0.022	0.065	0.985	0.000
11	0.034	0.014	0.010	0.951	0.015	0.000	-0.054	-0.064	-0.008	0.059	1.094	0.000
12	0.011	0.057	0.003	1.122	0.010	0.000	-0.107	-0.009	-0.011	-0.013	0.974	0.000
13	0.003	-0.107	-0.007	0.726	-0.011	0.000	0.288	0.036	0.079	0.077	1.334	0.000
14	0.361	-0.168	0.024	0.996	0.026	0.000	-0.614	0.018	0.067	-1.256	1.482	0.000
15	-0.102	-0.024	-0.005	1.029	-0.012	0.000	-0.237	0.109	0.069	0.874	1.366	0.000
16	0.010	-0.032	-0.003	0.900	-0.004	0.000	0.132	-0.025	0.037	-0.066	1.202	0.000

Table 3.6: Sad facial deformation table.

Region	a_1	a_2	a_3	a_4	a_5	a_6	b_1	b_2	b_3	b_4	b_5	b_6
1	-0.031	0.018	-0.010	0.916	0.002	0.000	0.008	0.009	-0.021	0.020	1.039	0.000
2	0.027	0.011	0.023	0.937	-0.031	0.000	0.037	0.000	-0.016	-0.070	1.034	0.000
3	-0.004	-0.505	0.135	1.185	-0.065	0.000	0.055	0.531	-0.305	-0.286	1.133	0.001
4	0.001	0.039	0.110	0.999	0.075	0.001	-0.020	0.021	-0.070	-0.033	0.965	-0.001
5	-0.001	-0.004	-0.074	0.996	-0.015	0.000	-0.026	-0.022	-0.111	0.039	0.930	0.000
6	-0.004	-0.003	0.011	0.979	0.015	0.000	-0.011	0.008	0.016	-0.041	1.051	0.000
7	0.013	0.009	-0.005	0.972	-0.014	0.000	-0.019	-0.008	-0.004	0.054	1.022	0.000
8	-0.071	0.013	0.044	1.006	0.025	0.001	0.025	-0.031	0.003	-0.024	1.009	0.000
9	-0.096	0.057	0.056	1.055	0.048	0.000	0.005	-0.007	0.005	0.004	1.009	0.000
10	-0.004	-0.008	0.011	0.951	0.040	0.000	-0.060	-0.012	0.051	-0.261	1.209	0.000
11	-0.032	-0.018	-0.007	1.045	0.008	0.000	-0.092	-0.022	0.040	0.288	1.219	0.000
12	-0.003	0.028	-0.002	1.046	-0.005	0.000	-0.028	0.020	0.003	0.037	1.004	0.000
13	0.010	-0.034	-0.002	0.933	0.000	0.000	-0.039	-0.015	0.042	-0.037	1.099	0.000
14	0.146	-0.305	0.087	0.252	0.255	0.000	0.184	-0.189	0.047	-0.223	1.083	0.000
15	-0.214	0.008	0.000	1.340	0.040	0.000	0.155	-0.022	-0.024	-0.466	0.848	0.000
16	0.003	-0.034	0.007	0.837	0.024	0.000	-0.043	0.085	-0.003	0.263	0.964	0.000

Table 3.7: Disgust facial deformation table.

Region	a_1	a_2	a_3	a_4	a_5	a_6	b_1	b_2	b_3	b_4	b_5	b_6
1	-0.028	0.017	-0.009	0.920	0.005	0.000	0.028	0.020	-0.014	0.031	1.018	-0.001
2	0.050	0.011	0.013	0.886	-0.016	0.000	0.024	-0.019	0.007	-0.018	0.986	0.000
3	0.050	0.417	0.332	0.736	-0.167	0.000	-0.179	-0.293	0.172	0.121	0.833	0.000
4	0.013	-0.020	0.166	1.017	0.031	0.001	-0.039	-0.004	-0.161	-0.078	0.789	0.001
5	-0.024	0.025	-0.273	1.030	-0.002	0.000	-0.055	0.012	-0.042	0.090	0.784	0.001
6	-0.016	0.007	0.020	0.949	0.069	0.000	-0.021	-0.020	-0.084	-0.034	0.771	0.000
7	0.030	0.003	-0.037	0.914	-0.085	0.000	-0.039	0.031	-0.109	0.050	0.689	0.000
8	0.029	-0.079	0.019	0.914	0.027	0.001	0.284	0.010	-0.053	0.021	0.964	0.002
9	-0.008	-0.522	-0.017	0.405	-0.021	0.000	0.477	0.264	-0.123	0.282	0.897	0.000
10	-0.053	0.028	0.007	0.868	0.100	0.000	-0.107	0.030	0.000	-0.261	1.095	0.000
11	0.045	0.008	0.007	0.894	-0.018	0.000	-0.069	-0.051	-0.031	0.081	0.970	0.000
12	-0.022	0.109	0.002	1.291	0.005	0.000	-0.062	0.027	-0.064	0.080	0.826	0.000
13	0.027	-0.093	0.004	0.829	0.018	0.000	0.190	0.052	0.049	0.102	1.198	0.000
14	0.249	-0.022	0.007	1.325	-0.004	0.000	-0.018	0.015	0.025	-0.182	1.146	0.000
15	0.064	-0.283	-0.091	-0.006	-0.311	0.000	0.190	-0.076	0.000	-0.371	1.032	0.000
16	0.105	-0.458	0.008	-0.471	0.038	0.000	-0.008	-0.189	0.028	-0.684	1.131	0.000

3.6 Automatic Facial Muscle Region Definition

The FDT defines the unique deformation characteristic for the sixteen facial regions that are described in Section 3.1. When an application domain uses the FDT parameters, there needs to be well defined boundaries for the sixteen facial regions. In this section, we describe a semi-automatic facial region grouping algorithm that uses the *Golden Ratio* or the divine proportion [65] of the face to generate the region boundaries. The algorithm requires the user to specify the coordinates of four facial feature points F_1, F_2, F_3, and F_4 as shown in Figure 3.12. Using those features, an automatic grouping of the sixteen facial regions is computed using the *Golden Ratio*. Table 3.8 defines the 21 boundary points for the left hand side of the face, while Figure 3.12 shows the corresponding locations of the 21 boundary points. We use the symmetry of the face to compute the boundary points of the right hand side of the face. The following list describes the variables and constants used in Table 3.8:

- a is the distance between the two eye pupils
- O is the midpoint between the eye pupils
- r is the radius of the eye, $r = \frac{a}{2\phi^3}$
- $\phi = 1.61803398874989$, which is the golden ratio [65]

Due to different facial structures and face shapes, it can be difficult get a clear approximation of the boundary points based on the golden ratio alone. To better approximate the facial regions, we allow for six editable feature points that can be modified to better fit region boundaries of the face shape.

These feature points are marked with a star ($\star$) in Table 3.8. Figure 3.13 shows an example of the facial regions being defined automatically using the golden ratio and the user defined facial feature points (F_1, F_2, F_3, and F_4).

Table 3.8: Boundary points definition in the $XY - plane$.

Point	X	Y
P_1	$O_x - \frac{a}{2}$	$O_y + a$
P_2	O_x	$O_y + a$
P_3	$O_x - a$	$O_y + \frac{a}{2}$
$\star P_4$	$O_x - \frac{a}{2\phi}$	$O_y + \frac{a}{2\phi}$
P_5	$O_x - a$	$O_y + r$
P_6	$O_x - \frac{a}{2} - r$	$O_y + r$
$\star P_7$	$O_x - \frac{a}{2} + r$	$O_y + r$
P_8	O_x	$O_y + r$
P_9	$O_x - \frac{a}{2}(1 + \phi)$	O_y
P_{10}	$O_x - \frac{a}{2}(1 - \phi)$	O_y
P_{11}	O_x	O_y
P_{12}	$O_x - \frac{a}{2}(1 - \phi)$	$O_y - r$
P_{13}	$O_x - \frac{a}{2} - r$	$O_y - r$
P_{14}	$O_x - \frac{a}{2} + r$	$O_y - r$
P_{15}	$O_x - a$	$F_3(y) + \frac{a}{\phi}$
$\star P_{16}$	$O_x - \frac{a}{2\phi}$	$F_3(y) + \frac{a}{\phi}$
$\star P_{17}$	$O_x - a$	$F_3(y) - \frac{a}{\phi^3}$
$\star P_{18}$	$O_x - \frac{a}{2\phi}$	$F_3(y) - \frac{a}{\phi^3}$
P_{19}	$O_x - \frac{a}{2}(1 - \phi)$	$F_3(y)$
P_{20}	$O_x - \frac{a\phi^2}{2}$	$F_4(y) - \frac{a}{\phi^2}$
$\star P_{21}$	$\frac{F_3(y)+F_4(y)}{2}$	$F_4(y) - a$

Point Inclusion Test

To be able to transform points within a facial region, we employ the axis aligned bounding boxes method and the point inclusion test. The enclosed muscle region is described as a polynomial shape, and it is used to identify if a point is located within the boundary area of the region. A point inclusion

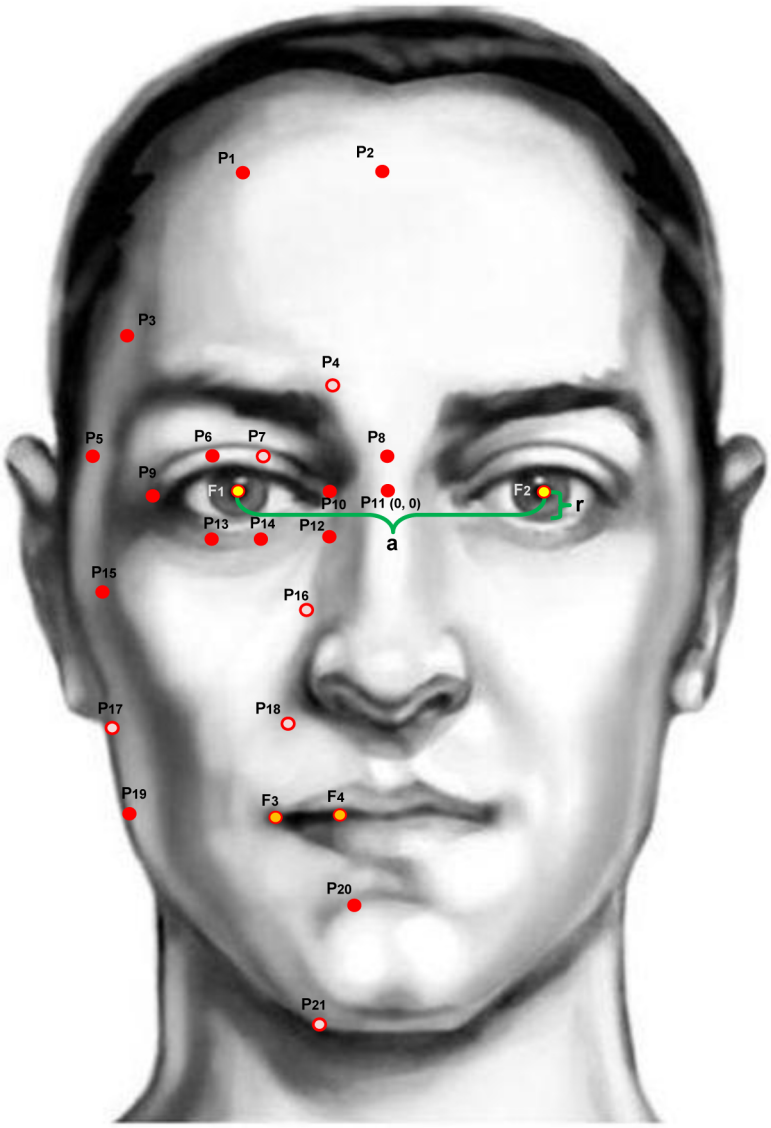


Figure 3.12: Automatically defined region boundary points. Facial illustration adopted, with permission, from [38].

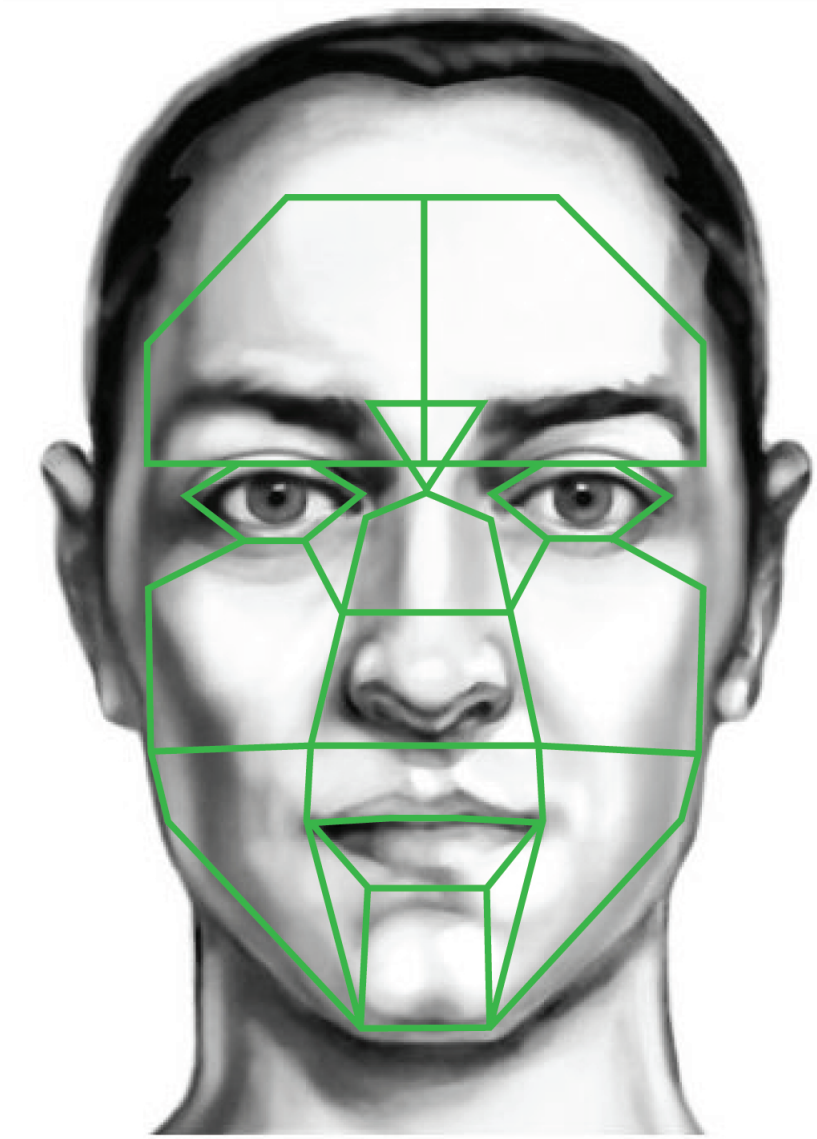


Figure 3.13: Facial regions defined automatically using the golden ratio. Facial illustration adopted, with permission, from [38].

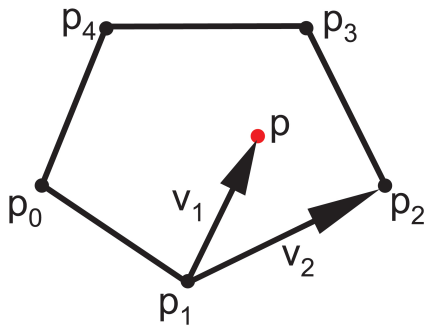


Figure 3.14: Illustration of the point inclusion test.

test is one of the simplest approaches to indentify if a point is located in the boundary area. Figure 3.14 illustrates how the inclusion test works, where v_1 and v_2 are the vector of (p_1, p) and (p_1, p_2), and the cross product of the vectors returns either a negative or positive value. If the overall cross product results for all the boundary points agree with the same signed value, then the point p is within the particular region. The computing time for a point inclusion test is very time consuming since for every point, it takes n iterations to compute, where n is the number of the corners of a convex shape. To enhance the calculation, the axis aligned bounding boxes is introduced. The bounding box stores the minimum and maximum corner positions of the polynomial muscle region. The program takes the 2D feature point and applies this method to determine if it is within the boundary. Then the specific point inclusion test is performed only if the point is found within the particular boundary region. This may be considered as a redundant process, but the majority of points may not bind within any of the defined muscle region. Therefore, it reduces the number of performance calls for the point inclusion test which is a time-consuming process.

Chapter IV

Facial Expression Recognition using Quadratic Deformation Models

This chapter describes a novel approach for recognizing facial expressions based on Active Appearance Models (AAM) for facial feature tracking and the quadratic deformation model representations of facial expressions described in Chapter 3. Thirty seven facial feature points are tracked based on the MPEG-4 Facial Animation Parameters layout. The proposed approach relies on the Euclidean distance measures between the tracked feature points and the reference deformed facial feature points of the six main expressions (surprise, fear, anger, sadness, disgust, and smile). An evaluation of 30 model subjects, selected randomly from the Cohn-Kanade Database [56], was carried out. Results show that the main six facial expressions can successfully be recognized with an overall recognition accuracy of 89%. The proposed approach yields promising recognition rates and can be used in real time applications.

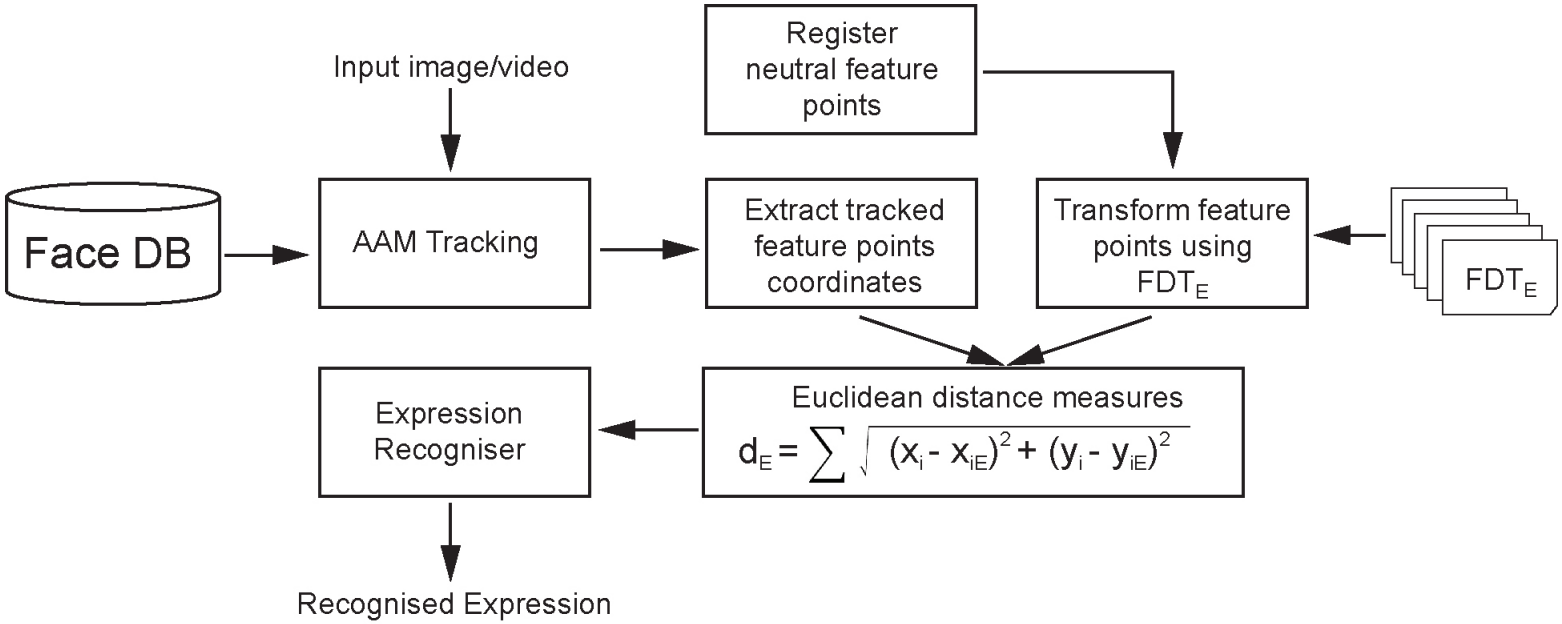


Figure 4.1: Facial expression recognition process.

4.1 Classification of Facial Expressions

In this section, we demonstrate how to utilize the FDT_E in the recognition of the six main expressions. Figure 4.1 shows the recognition process which consists of five main steps: (1) tracking feature points, (2) registering the feature points of the neutral expression, (3) using the FDT_E to transform the feature points from step 2, (4) using the Euclidean distance for similarity measurements, and (5) recognizing the facial expression. The following sections describe the above steps in more details.

4.1.1 Step (1): Tracking Feature Points

The initial locations of the tracked points are based on the MPEG-4 Facial Animation Parameters (FAPs) [72]. The complete set consists of 68 FAPs, but in this study a sub-set of 37 FAPs is used since many points were considered irrelevant. In particular, we omit any FAPs that do not get involved directly in driving the facial expressions. For example, this includes the "high-level" FAPs (viseme and expression), tongue FAPs, and FAPs dealing with ears and global head rotation.

Using the Active Appearance Model (AAM) tracking algorithm [89], we tracked the movement of the 37 facial feature points as shown in Figure 4.2. The tracking process is carried out in two main phases:

- **Training Phase:** The training phase generates a trained model that can be used for extracting the desired facial features. The facial expression images of 30 subjects, acquired from the Cohn-Kanade Database [56], were annotated using the AAM annotation software tools provided by Tim Cootes [20]. Figure 4.4 shows some examples of the images used

in the training phase.

- **AAM Fitting:** The annotated set of images are then used as an input to the AAM trainer to produce a trained model capable of locating the desired features. AAM Fitting is described in more detail in Appendix A. Figure 4.3 gives an example of the facial feature points tracking of the surprise facial expression.

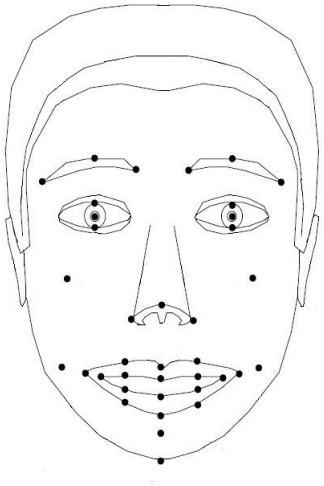


Figure 4.2: The tracked points based on the MPEG-4 FAPs layout.

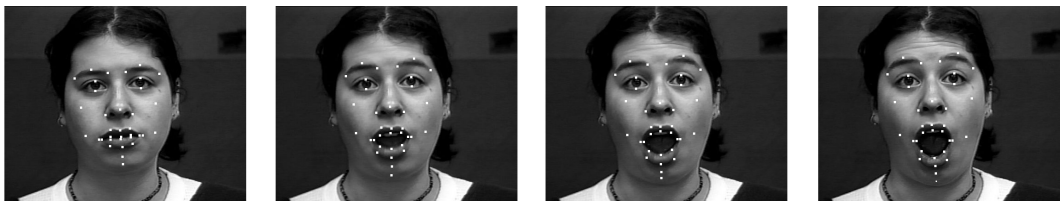


Figure 4.3: An example showing the tracking of facial feature points from selected frames of the surprise facial expression. Images obtained from the Cohn-Kanade facial expression database (© Jeffrey Cohn) [56].

To analyze the details of the facial feature points, global head movements are eliminated by pose and scale normalisations. The normalisation is done

based on two global facial locations (eye pupils) p_1 and p_2. The normalisation process is the same as the one described in Section 3.4.2 of the previous chapter.

4.1.2 Step (2): Registering the Neutral Expression and using the FDT_E

The normalized facial feature point locations are registered for the neutral expression at the start of the recognition session. The registered points for the neutral expression are then used as a reference (R_{neutral}) for the duration of the session.

4.1.3 Step (3): using the FDT_E

In this step, we apply the FDT_E deformation parameters (described in Section 3.4.3) to the reference points R_{neutral} from step (2). This will formulate a reference for each of the expressions (R_E, where E = <surprise, fear, anger, sadness, disgust, and smile>). Using the set of deformed references, we can recognize the facial expression based on similarity measures as explained in step (4).

4.1.4 Step (4): Euclidean Distance Similarity Measure

After having extracted the tracked facial feature points from step (1) and using the set of deformed expression references (R_E) from step (3), our next task is to find the similarity measure d such that $d(P_i^T, P_i^E)$ is small if and only if P_i^T and P_i^E are close, where in this case P_i^T represents the tracked points and P_i^E represents the deformed reference points as shown in Figure 4.5.

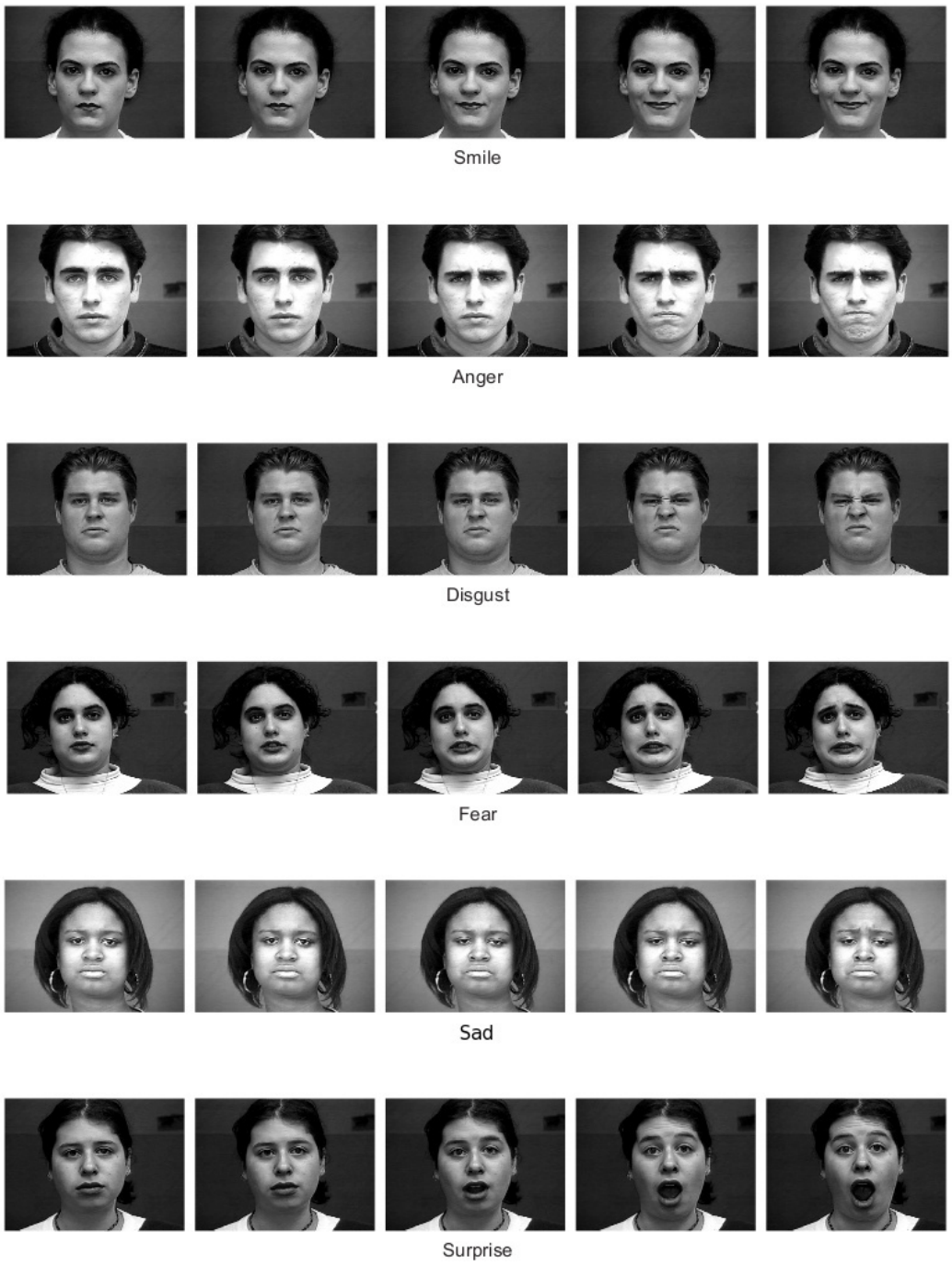


Figure 4.4: Example of the six primary facial expressions obtained from the Cohn-Kanade facial expression database (© Jeffrey Cohn) [56].

In our method, we compute d_E by summing the Euclidean distances between the tracked points and R_E as follows:

$$d_E = \sum_{i=1}^{n} \sqrt{(x_i^T - x_i^E)^2 + (y_i^T - y_i^E)^2} \tag{4.1}$$

where n is the total number of facial feature points and d_E is the total sum of distances for expression E. Since E is a set of the six main expressions, we compute a vector of six Euclidean distance measures, which will be used in Step (5) for recognizing the facial expression.

4.1.5 Step (5): Facial Expression Recognition

Using the Euclidean distance measures' vector, computed from step (4), facial expressions are classified based on the following rule [37]: "A small distance is equivalent to a large similarity." Therefore, the smallest value computed in the Euclidean distance vector is chosen to be the recognized expression.

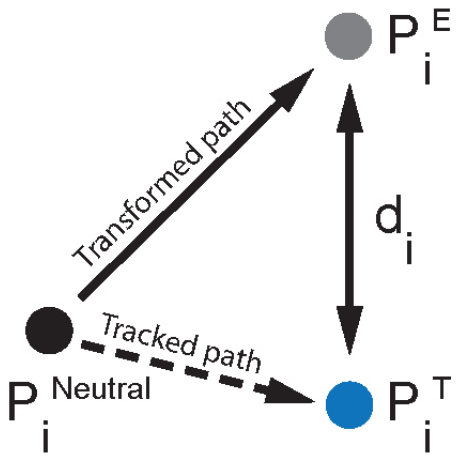


Figure 4.5: Euclidean distance measures.

4.2 Experimental Results and Discussion

To evaluate our facial expression recognition method in terms of accuracy, we have preformed experiments on 30 subjects[1] (15 females and 15 males) that were chosen randomly from the Cohn-Kanade Database (CKDb) [57]. Using the CKDb to randomly select facial expression images allowed us to ensure the diversity of the chosen images in terms of facial characteristics such as shape, size and texture. Each subject expressed the six main facial expressions from the neutral facial expression state.

Overall, the selected 30 subjects database consists of 992 natural images and 2432 images with facial expressions, of which there are, 448 images for anger, 296 images for disgust, 346 images for fear, 532 images for smiling, 423 images for sadness and 387 images for surprise. For each subject, one neutral image and all expressed images were used in the evaluation experiment.

Using the classification method, described in Section 4.1.5, we performed the evaluation experiments for each subject. The overall recognition accuracy results are summarized in Table 4.1.

Table 4.1: Confusion matrix for facial expression recognition.

Expression	**Smile**	**Anger**	**Sad**	**Surprise**	**Fear**	**Disgust**
Smile	**93.3%**	0.0%	0.0%	0.0%	0.0%	6.7%
Anger	0.0%	**93.3**%	6.7%	0.0%	0.0%	0.0%
Sad	0.0%	3.3%	**96.7%**	0.0%	0.0%	0.0%
Surprise	0.0%	0.0%	0.0%	**100.0%**	0.0%	0.0%
Fear	0.0%	3.3%	0.0%	6.7%	**66.7%**	23.3%
Disgust	3.3%	13.3%	0.0%	0.0%	0.0%	**83.3%**

[1] Note: subjects used in the training phase of the tracking process are omitted from selection in the experimental evaluations.

The results show an 88.9% overall recognition accuracy of all expressions with the highest recognition rates of surprise (100%) and the lowest recognition rates of fear (66.7%).

The distinct feature of the 'surprise expression' (opening the mouth and dropping the jaw) allows the system to have a 100% recognition accuracy rate of the surprise expression. While, the deformation features of the fear expression (pulling the mouth corners and a slight opening of the mouth) can overlap with other expressions (such as disgust), which can be one reason to explains its lower recognition rate of 66.7%. However, we further investigate (in Section 4.3) the database used in obtaining the FDT parameters and the CKDb to determine the cause of the lower recognition rate. Moreover, one way to improve recognition rates of facial expressions that deform in a similar manner is to compute similarity measures based on distinct features of different facial expressions by assigning different weight values. This will be explored as a future extension to this work.

4.3 Data Evaluation

In this section we look more deeply into evaluating the database used to extract the quadratic deformation models. Moreover, we further evaluate the facial expression recognition system by evaluating the Cohen-Kanade facial expression database.

4.3.1 Facial Expression Data for the Quadratic Deformation Model

To analyse and understand the proposed deformation models, an evaluation of the data used (Figure 4.6 shows an example of a facial expression

data set) to collect the deformation parameters was performed. Out of the twelve expression photographic sets (that were used to obtain the deformation parameters) we used ten sets of subjects who allowed us to evaluate their images. Each set contained the images of a subject performing the six main expressions (surprise, fear, anger, sadness, disgust, and smile).

Figure 4.6: An example from the facial expression database used to obtain the quadratic deformation tables. The example shows an actor performing the six primary expression. Images included with permission.

The participants' task was to look at the expression images and to define what facial expressions they saw. They were asked to choose what facial expression was in each image by selecting one label from the following: (surprise, fear, anger, sadness, disgust, and smile). Images appeared in a random order to ensure that they were not displayed in a repeated pattern. The results of twenty participants' evaluations of the facial expression datasets are summarised in Table 4.2.

The results in Table 4.2 show that on average the facial expression in the images can be recognised with an accuracy of 71.8%. All participants identified the smile expression easily. However, the fear expression was poorly identified with an accuracy of 38%, while 50% thought it was the surprise expression. The confusion can be explained by the fact that there are some FACS [30] Action Units that coincide in both the surprise and fear expressions. The data images of facial expressions originate from acted expressions

Table 4.2: Confusion matrix for the facial expression data.

Expression	Smile	Anger	Sad	Surprise	Fear	Disgust
Smile	**100%**	0.0%	0.0%	0.0%	0.0%	0.0
Anger	1.5%	**71%**	12%	0.0%	8.5%	7.0%
Sad	1.5%	10.5%	**80.5%**	1.5%	1.5%	4.5%
Surprise	1.5%	0.0%	0.0%	**77.5%**	21%	0.0%
Fear	0.0%	3.0%	9.0%	50.0%	**38%**	0.0%
Disgust	0.0%	32%	3.0%	0.0%	1.5%	**63.5%**

and not natural expressions, which can largely influence the AU coincided facial expressions [111]. Anger was identified with an accuracy of 71%, while it was slightly confused with its AU coincided facial expressions sadness and disgust. Moreover, disgust was recognised with an accuracy of 63.5% and was confused by 32% with anger. Similarly, the surprised and sadness expressions were accurately identified with 77.5% and 80.5% respectively. Surprised was confused by 21% with fear, while sadness was confused by 10.5% with anger.

Overall the results show a high accuracy of identification of the acted facial expressions data images, which reflects on the accuracy of the quadratic deformation parameters collected from the evaluated data.

4.3.2 Cohn-Kanade Database for Facial Expression Recognition

The facial expression recognition system described in this Chapter was tested using the facial expressions of 30 subjects chosen randomly from the Cohn-Kanade Database facial expression database (CKDb) [57] (Examples of facial expressions acquired from CKDb are shown in Figure 4.4). To further understand the results obtained by the facial recognition system, we asked participants to evaluate the facial expressions of 20 subjects chosen randomly

Table 4.3: Confusion matrix of the Cohen-Kande facial expression database used in the experiment.

Expression	**Smile**	**Anger**	**Sad**	**Surprise**	**Fear**	**Disgust**
Smile	**98.5%**	0.5%	0.0%	0.0%	1.0%	0.0%
Anger	0.0%	**76.5%**	5.5%	3.5%	4.5%	10%
Sad	0.0%	4.5%	**84.5%**	5.0%	2.5%	3.5%
Surprise	0.0%	0.5%	0.0%	**89.5%**	10%	0.0%
Fear	8.0%	0.5%	0.5%	8.5%	**37.5%**	45%
Disgust	0.0%	26%	3.0%	1.0%	1.5%	**68.5%**

from the 30 subjects used during testing.

For each video, participants saw one specific facial expression and they are asked to choose what facial expression they saw by selecting one label from the following: surprise, fear, anger, sadness, disgust, and smile. They are asked to judge the whole sequence as one facial expression and not to pay attention to the quality of the video. The results of twenty participants, who carried out, the evaluation are summarised in Table 4.3.

The results in Table 4.3 show that the majority of participants found that the smile expression was easy to identify with a rate of 98.5%. Anger was correctly identified with a rate of 76.5%, but 10% of the time it was mistaken for disgust. Similarly, the accuracy rate of identifying surprise was 89.5% and it was mistaken by 10% for fear. On the other hand, fear was poorly distinguished with an accuracy rate of 37.5%, as it was recognised as disgust for 45% of the time. Disgust was classified correctly by participants with an accuracy of 68.5% and it was mistaken with anger for 26% of the time.

The evaluation results obtained in this section allows us to further un-

Table 4.4: Evaluating the accuracy performance of the facial recognition system.

Expression	**Recognition System** Table 4.1	**FDT database** Table 4.2	**Cohn-Kanade** Table 4.3
Smile	93.3%	100%	98.5%
Anger	93.3%	71.0%	76.5%
Sad	96.7%	80.5%	84.5%
Surprise	100.0%	77.5%	89.5%
Fear	66.7%	38.0%	37.5%
Disgust	83.3%	63.5%	68.5%

derstand the performance of the facial expression recognition system. Table 4.4 compares the classification rates of Table 4.1, Table 4.2 and Table 4.3.

From Table 4.4, we can see that the recognition system's accuracy to classify facial expressions is based on the data collected and how well the expressions are acted. In all cases, except fear, facial expressions where acted similarly in both the quadratic deformation facial expression data and the Cohen-Kanade database. Therefore, this resulted in high classification accuracy from only 37 tracked facial feature points. In the case of fear, we determined that the way actors showed fear in the images of the quadratic deformation facial expression data was different to the acted fear in the Cohen-Kanade database. However, it can be seen that the human recognition of the fear expression was poorly done in both the FDT and Cohen-Kanade databases with a rate of 38% and 37.5% respectively. The reason for this can be derived from the actors not-knowing how to show or perform the fear expression. This is also supported by the work of Wimmer and Zucker [111] [115], who showed in a study on the Cohen-Kanade database, that poor recognition results by humans come from using performed expressions rather

than natural expressions. Therefore, the poor acting of the expression can influence the recognition rate of the described system and introduce poor results in classifying the expression with lower rates such as the expression fear (66.7%).

Chapter V

Expressive Caricatures using Quadratic Deformation Models

This chapter describes a new approach to generating facial caricatures from an input face image. Our approach allows the user to control the exaggeration level of the facial expression imposed on the caricature drawing. It is also capable of producing an expressive facial animation of the caricature drawing as shown in Figure 5.13. The three main components of our approach are: (1) A quadratic deformation model for the transformation of feature points, which can effectively map any of the six main expressions (surprise, fear, anger, sadness, disgust, and smile) to a facial image, (2) an interpolation method for manipulating the face's appearance and expressivity of the caricature, and (3) an automatic moment-based stroke rendering algorithm to render extracted facial features.

5.1 System Overview

We now briefly describe the main components of our caricature generation system, as illustrated in Figure 5.1. The developed system employs a combination of feature extraction algorithms, facial expression representations

using quadratic deformation models, and a stroke-based caricature rendering algorithm.

The aim is to create an expressive caricature from a given neutral face image. The first step is to extract features from the image that will undergo the deformations. AAM (described in Appendix A) and digital matting [94] are used to identify facial features of interest, such as eyebrows, eyes, nose, lips, ears and hair. We call this part, rendering path extraction; as the extracted path will serve as the painting line for the caricature rendering algorithm.

Once the features have been extracted, they are transformed using the quadratic deformation model representations of facial expressions (described in Chapter 3). Finally, the caricature rendering algorithm overlays strokes on top of the extracted path to produce a caricature that appears to be a rendering of the original image. However, in the output image the subject's expressions have been noticeably altered or exaggerated. The following sections explain each of the main parts of our method in more detail.

5.2 Rendering Path Extraction

The rendering path represents the location of the features where the strokes will be painted to create the final caricature's appearance. The Non-Photorealistic Rendering (NPR) algorithm, described in Section 5.4, takes as an input, the rendering path and generates the final caricature appearance. The rendering path for the hair and the facial features are generated independently. To extract the rendering path of the hair[1], we use digital matting and edge

[1] The shape of the hair also includes the shape of the ears.

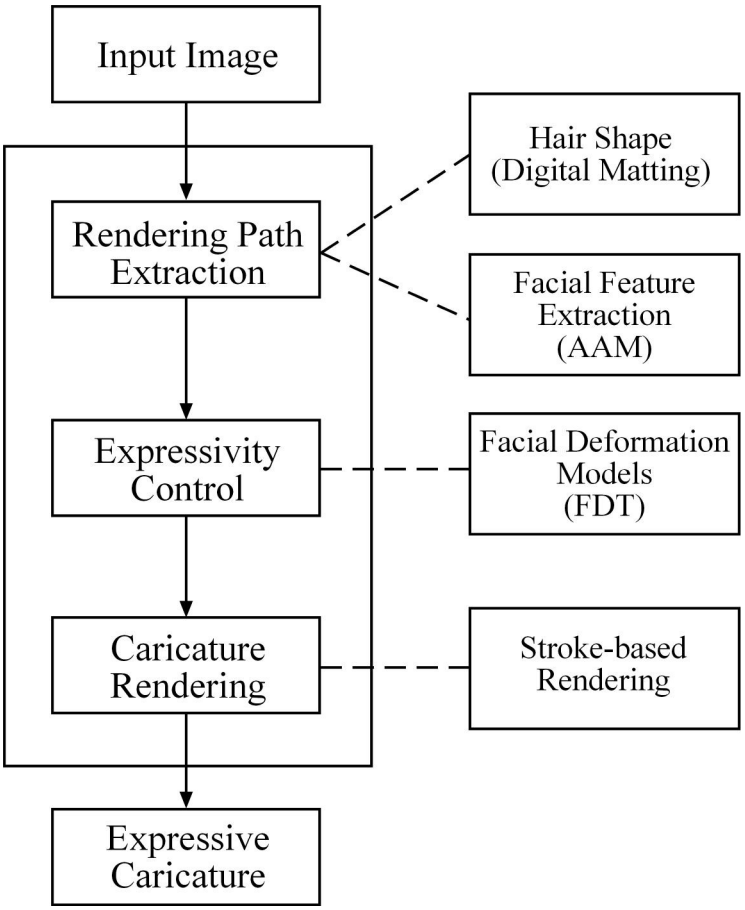


Figure 5.1: The caricature system components.

detection algorithms, while AAM facial feature extraction is used to determined the strokes rendering path for the facial parts. The following sections describe the process in more detail.

Figure 5.2: Hair and ear shape segmentation. (a) Marked region of interest, (b) digital matting segmentation, (c) threshold image and (d) edge detection (rendering path).

5.2.1 Hair and Ears Shape

The shape of the hair can vary from one individual to another and for that reason it is processed independently from the facial feature parts. To segment the hair from the face and background regions, we use the digital matting algorithm of Sindeyev and Konushin [94]. The method is first initialized by roughly marking out the region of interest (i.e. hair). The digital matting tool[2] provided by Sindeyev and Konushin [94] is then used to segment the hair region out. The edges of the hair region (detected using the canny algorithm [13]) serve as the strokes' rendering path. Figure 5.2 illustrates the general process.

[2] `http://graphics.cs.msu.ru/en/science/research/imageprocessing/matting`

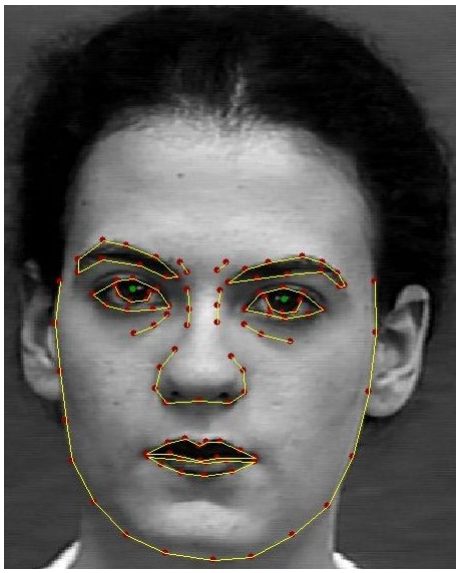


Figure 5.3: Annotations of facial features. Image adopted, with permission, from [57].

5.2.2 Facial Feature Extraction

The rendering paths for the facial parts are generated by extracting the shapes using AAM. This process is carried out in two main phases: a training phase and an AAM fitting phase.

Training Phase: The training phase generates a model that can be used for extracting the desired facial features. The facial images of the neutral expression of 30 subjects, acquired from the Cohn-Kanade Database [56], are annotated using the AAM annotation software tools provided by Tim Cootes [20]. Figure 5.3 illustrates the annotated features of interest for the purpose of caricature rendering (eye brow features, eyes features, nose, lips, and face).

AAM Fitting Phase: The annotated set of images are then used as an input to the AAM trainer to produce a trained model capable of locating the desired features. In our system we use the method of Saragih and Goecke

[88] [89] (described in Appendix A). Figure 5.4 demonstrates an example of the facial feature extraction using AAM.

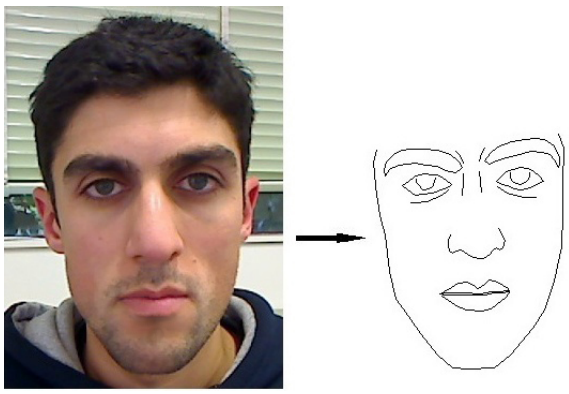


Figure 5.4: Extracting facial features using AAM.

5.3 Expressivity of the Facial Caricature

Once the features have been extracted, they are transformed using the facial expression representations described in Chapter 3. In our approach, we map the defined muscle based facial regions into the corresponding extracted facial feature lines. This allows us to generate different deformations and exaggerations of the extracted facial features and form the desired caricature appearance. The following sections describe in more detail how the expressivity of the facial features is controlled.

5.3.1 Controlling Exaggeration Levels

The FDT parameters derived using the quadratic deformation model (described in Section 3.4.3) can be applied on any feature line of the caricature by first identifying the region to which the line belongs, and then applying the

corresponding transformation parameters to every point on that line. Moreover, the way the FDTs are defined, allows us to control the expressiveness of a caricature and its level of exaggeration. This is done by interpolating the FDTs parameter values between the neutral expression and the desired expression. Extrapolating the parameters beyond the desired expression will generate an exaggeration of the appearance of the facial parts. In this context, Equations 3.3 and 3.4 can be re-written as follows:

$$x'_{i(E)} = \sum_{u=0}^{2}\sum_{v=0}^{2} A_{uv(E)} x_i^u y_i^v \tag{5.1}$$

$$y'_{i(E)} = \sum_{u=0}^{2}\sum_{v=0}^{2} B_{uv(E)} x_i^u y_i^v \tag{5.2}$$

where suffix (E) denotes a transformation specific to an expression E, and $A_{00} = a_6$, $A_{01} = a_5$, $A_{10} = a_4$, $A_{02} = a_3$, $A_{11} = a_2$, $A_{20} = a_1$, with similar mapping for coefficients b_i.

A parametric linear interpolation between a neutral expression N, and a given expression E is then given by

$$\begin{aligned} x'_i &= (1-t)x'_{i(N)} + t x'_{i(E)} \\ &= \sum_{u=0}^{2}\sum_{v=0}^{2}((1-t)A_{uv(N)} + tA_{uv(E)})x_i^u y_i^v \end{aligned} \tag{5.3}$$

$$\begin{aligned} y'_i &= (1-t)y'_{i(N)} + t y'_{i(E)} \\ &= \sum_{u=0}^{2}\sum_{v=0}^{2}((1-t)B_{uv(N)} + tB_{uv(E)})x_i^u y_i^v \end{aligned} \tag{5.4}$$

$$0 \leq t \leq 1$$

The parameter t defines the degree of exaggeration of the expression E, used for producing the caricature. The value of t could be increased above 1 to extrapolate the coefficients beyond the normal values for that expression. Our system allows the user to interpolate the parameter t by varying the value between $0 \leq t \leq 1$ or extrapolate the value $t > 1$ (as demonstrated in Figure 5.5).

A continuous variation of the parameter t produces an animation sequence for the given facial expression. An example of a caricature animation for the smile expression is shown in Figure 5.13.

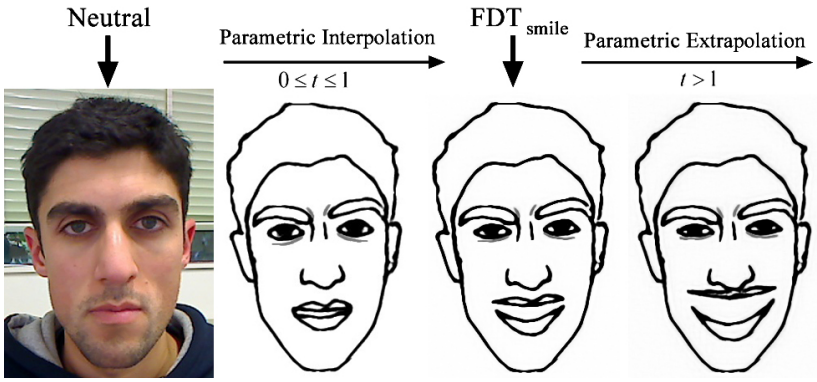


Figure 5.5: Caricature exaggeration levels of the 'smile' facial expression.

5.4 Caricature Rendering

To enhance the appearance of the generated caricature, we employ a stroke-based non-photorealistic rendering algorithm. Some of the concepts of the algorithm are derived from my masters work [69], as described in this section. However, several enhancements have been made for the purpose of stylized caricature rendering.

Initially, the algorithm starts with a blank image (canvas), and then builds a composition of the caricature appearance, by progressively applying strokes along the rendering path, generated from section 5.2.2. Three main steps are involved in the preparation and composition of the caricature: (1) define the stroke locations, (2) compute stroke attributes, and (3) compose the strokes. The following sections explain each of the steps in more detail.

5.4.1 Strokes' Locations Image

Every pixel along the strokes' rendering path, generated from section 5.2.2, serves as a reference to a stroke position along the path. To intensify the non-photorealistic effect of the final caricature composition, we randomly add extra stroke positions for every pixel along the rendering path. Figure 5.6 demonstrates how a pixel location (along the path) is used to generate a random number of new stroke locations around it. The generated stroke locations are stored in a buffer as coordinates (x, y). The stroke locations' buffer is then used in later phases to compute and map the strokes into their position. An example image of the stroke locations' buffer is shown in Figure 5.7.

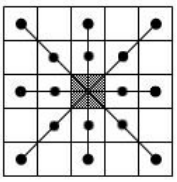


Figure 5.6: Possible points to be randomly selected as a stroke location.

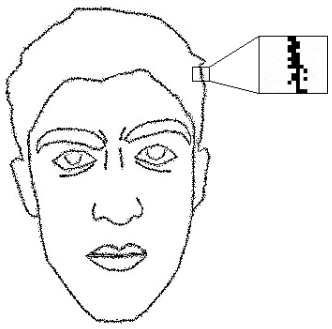


Figure 5.7: Randomly generated stroke locations around the strokes' rendering path.

5.4.2 Stroke Attributes

In our approach, strokes are represented as rectangular approximated shapes, as shown in Figure 5.8. Each of the strokes has the following attributes: colour, position, orientation, and size. The colour of the strokes are based on the extracted rendering path, i.e., for the main facial features the stroke colour is black and for extra facial marks we use grey. The position of the stroke is acquired from the strokes' locations buffer (Section 5.4.1). We apply the geometric moment shape descriptors, along the stroke rendering path, to determine the stroke's orientation θ and height h, while the width w of the stroke is a value determined by the user (this is because the width of extracted path is small). Appendix B describes in details how geometric moments are used to compute the stroke attributes.

5.4.3 Caricature Composition

To render strokes along the drawing path, we use a coarse-to-fine rendering approach. Larger strokes are placed first along the strokes' rendering path

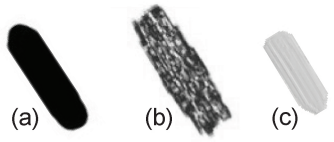


Figure 5.8: Different types of strokes that can be used for painterly rendering. (a) Colour pen stroke template. (b) Brushed stroke used in my masters work [69]. (c) Brush stroke used by Shiraishi and Yamaguchi's algorithm [92]

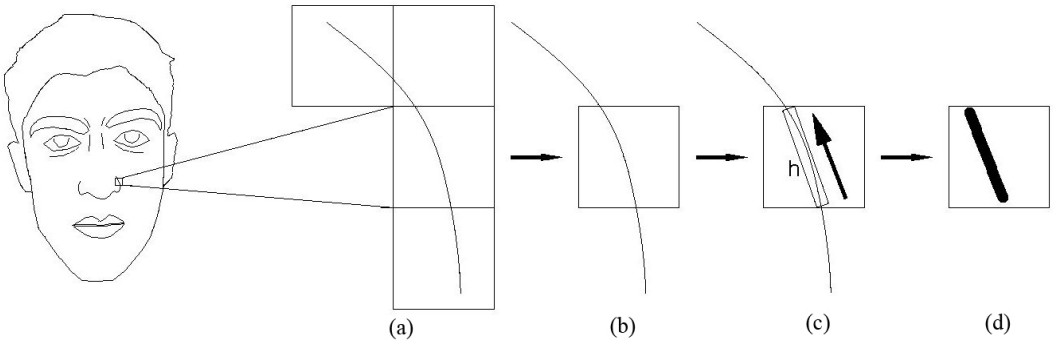


Figure 5.9: The cropping process along the strokes rendering path. (a) Shows a segment of the path subdivided into $s \times s$ windows, (b) cropping the window to be processed, (c) applying geometric moments to find the stroke attributes and (d) the initial stroke is painted along the path.

followed by smaller strokes around the rendering path. This ensures a complete rendering of the caricature with extra artistic effects. The following steps explain how the strokes attributes are computed:

Large strokes: We subdivide the extracted rendering path image into $(s \times s)$ windows, where s is the size of the window. Each segment is cropped, and then used to compute the stroke attributes for that part of the drawing path. Figure 5.9 demonstrates the process.

Small strokes: The image is further subdivided into $(s/2 \times s/2)$ window segments for every extra location stored in the stroke locations buffer. Coordinates of the stroke locations serve as the centre point of the subdivided window segments. Each segment is cropped and used to compute the stroke attributes using geometric moment functions. The central position of the stroke is the same as the location coordinates stored in the stroke locations buffer.

After determining the attributes of all the subdivided window segments, the strokes are rendered into their positions. The strokes are successively rendered one by one, using alpha blending, on a white canvas. Finally, the composition process is complete when the inner regions of the eye pupils are rendered using a flood-fill algorithm with a black colour.

5.5 Caricature Examples

Figures 5.11 and 5.12 illustrate examples of expressive caricatures generated using our proposed approach. The figures shows a neutral facial input image used to generate expressive caricatures of the facial expressions neutral, surprise, fear, anger, sadness, disgust, and smile. The expressiveness of the

caricature is controlled using the FDT's parameter values, while preserving the facial characteristics of the input image.

Figures 5.13 and 5.14 show examples of caricature animation for the 'smile' facial expression. Figure 5.10 shows more results of selected expressive caricatures from different individuals.

The illustrated examples show stylized renderings of caricatures from a given image of a face, with the ability to map any one of the main six expressions and control the degree of its expressiveness on the generated caricature.

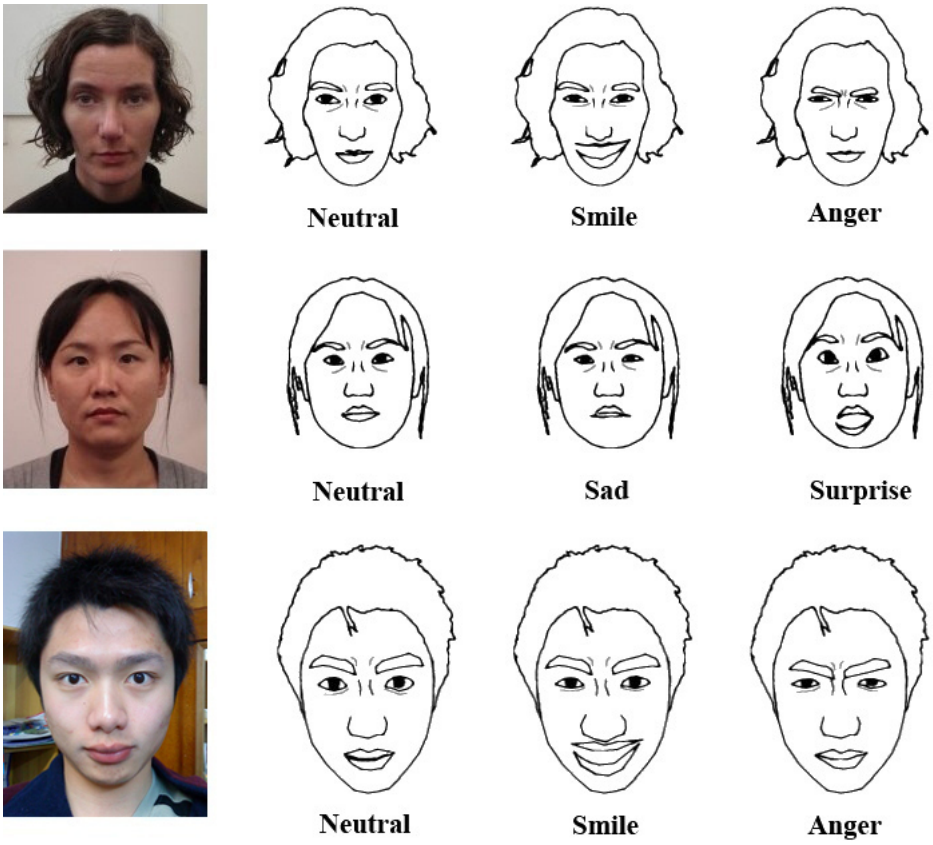


Figure 5.10: Selected expressive caricatures generated by the system. Images included with permission from subjects.

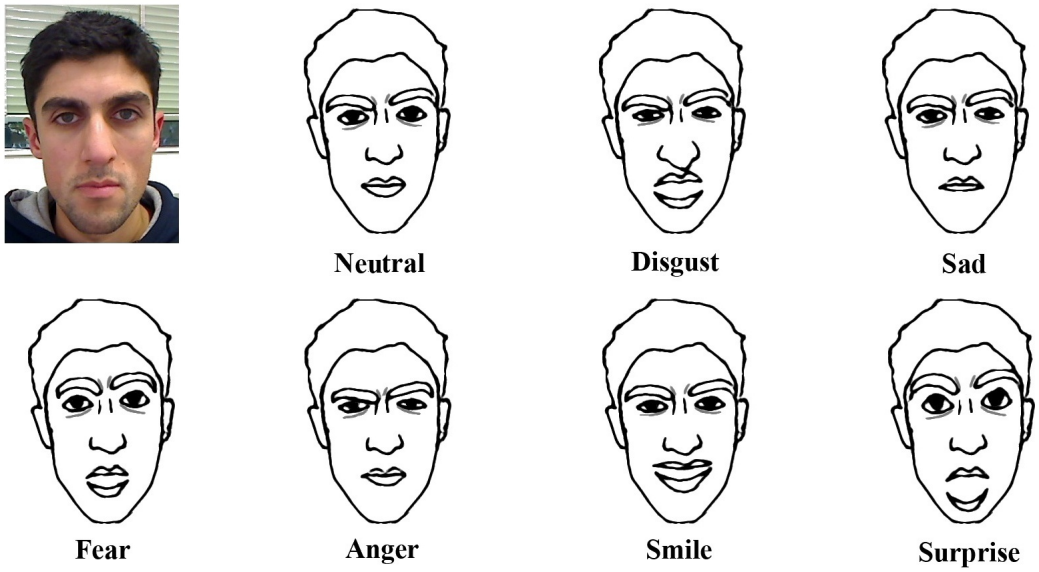


Figure 5.11: An example of expressive caricatures generated from an input image.

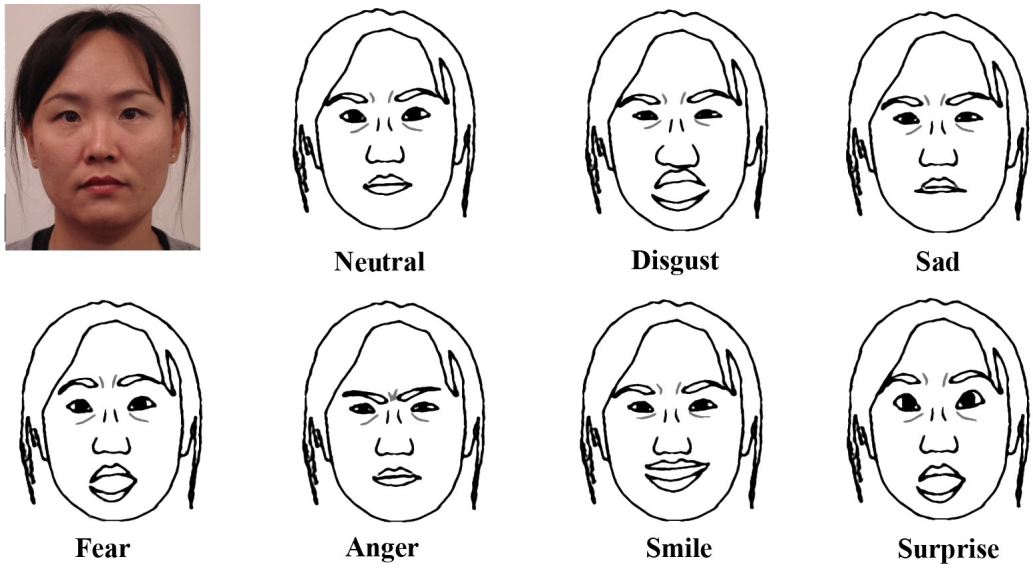


Figure 5.12: Another example of expressive caricatures generated from an input face image.

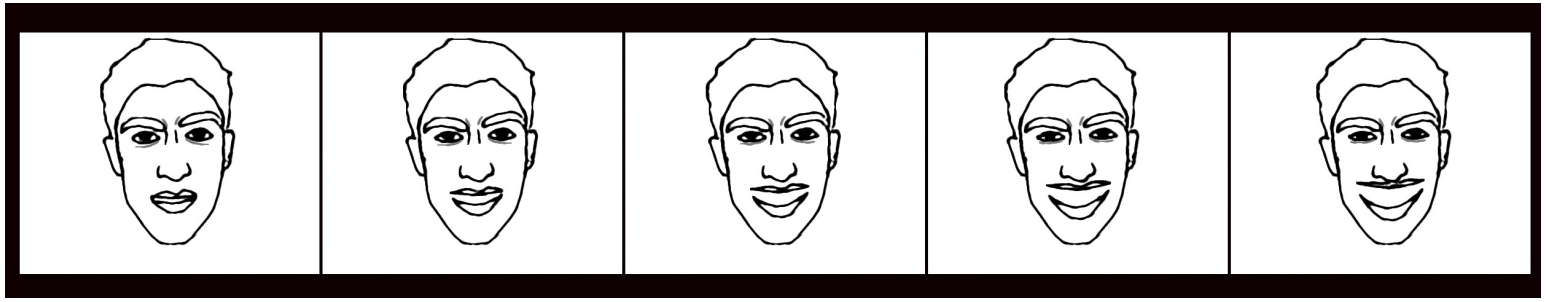


Figure 5.13: An animation sequence of a caricature with a smile facial expression.

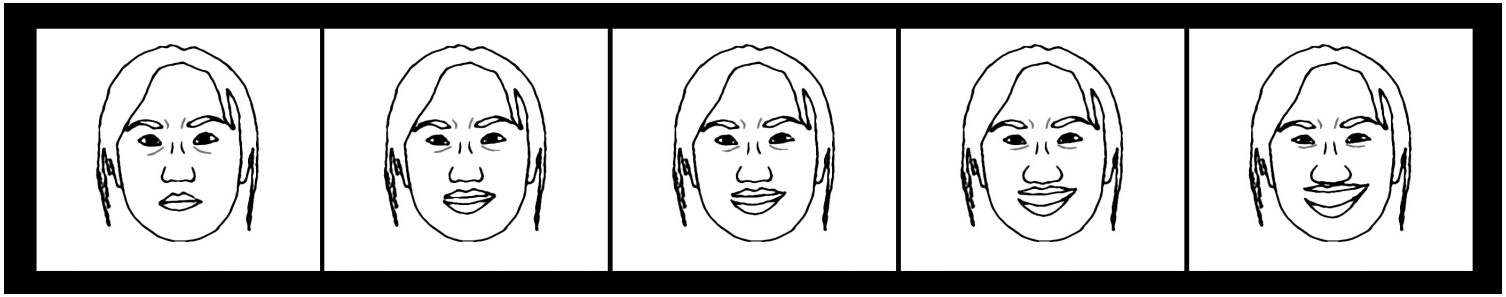


Figure 5.14: A sequence of caricature animation of the smile facial expression.

5.6 Caricature Algorithms: Comparative Evaluations

Computer generated caricatures are part of the Non-Photorealistic Rendering (NPR) research field, and evaluating NPR is a challenging problem. A few researchers, such as Gooch *et al.* [43], have used psychophysical studies to assess how their generated caricatures can influence the speed of recognising and learning tasks. However, Hertzmann [44] described NPR evaluations as:

> One challenge of NPR research is that it is difficult to perform meaningful scientific tests in a field where judgments are ultimately subjective.
> ...
> The difficulty of defining a scientific methodology does not make the work any less valuable, although the evaluation is ultimately somewhat subjective. Like all research, NPR requires a great deal of guess work and stumbling around in the dark; only after experience does the community gain some feeling for which ideas are valuable. One can easily find examples of useful reasoning and research areas where exact quantitative measurements are impossible (e.g. in political science), and, conversely, quackery justified by plausible scientific methodology (e.g. in political science).

In this section, we evaluate the expressive caricature algorithm described by performing a comparative analysis with similar caricature generator algorithms. Each of the algorithms are tested on the support of the following components:

1. **Active Appearance Model (AAM)/Active Shape Model (ASM):** Does the method use AAM or ASM to locate or extract facial features?

2. **Image segmentation:** Are image segmentation algorithms used in the process of caricature generation?

3. **Artist prototypes:** Does the system require the use of real artist's drawings as part of the learning process of the caricature generator?

4. **Real facial muscle deformations:** Can the system simulate a real muscle deformations to generate different facial appearances?

5. **Brush-stroke rendering:** Is a stroke rendering approach used to achieve the stylized rendering of the caricature?

6. **Exaggeration:** Does the system support the exaggeration of the facial features?

7. **Facial expressions:** Can the system resemble different facial expressions?

Table 5.1 shows the comparative analysis of different components and properties supported by the approach proposed in this thesis and other similar algorithms. The comparative analysis results show that the main advantage of the proposed caricature algorithm is the use of real muscle deformations to change the facial appearance and generate the main facial expressions. The algorithm proposed by Su *et al.* [96] also generates expressive caricatures by mapping the six main facial expressions; however, their method uses vector-muscle based models which can be a time consuming algorithm. Most methods, use AAM or ASM to extract the shape of the main facial feature parts (eyes, eyebrows, lips, and nose), which in turn allows for

the ease of exaggerating these parts. Furthermore, most methods use thresholding and line-template approaches to generate their stylised rendering of the caricatures, while, our method uses a stroke-based rendering to generate a stylised artistic style.

	AAM/ASM	Image segmentation	Artist's prototypes	Real muscle deformations	Stroke rendering	Exaggeration	Expressions
Our approach	√	√		√	√	√	√
Chen *et al.* [17, 18]	√	√	√			√	
Chiang *et al.* [19]		√	√			√	
Chen *et al.* [16]	√	√	√				
Tseng *et al.* [101]	√					√	
Mo *et al.* [66]	√					√	
Su *et al.* [96]	√	√				√	√

Table 5.1: Comparative analysis of the different facial caricature algorithms.

Chapter VI

Facial Animation using Quadratic Deformation Models

In this chapter we propose a mechanism to synthesize facial expressions on 3D faces compliant with the MPEG-4 facial animation standard. We demonstrate the use of FDT parameters to deform the face of a 3D character by combining the MPEG-4 facial animation standard with the quadratic deformation model representation of facial expressions (described in Chapter 3). To illustrate our proposed approach we use the Greta Embodied Conversational Agent (ECA) [79] to generate different expressive states of the agent.

6.1 MPEG-4 Facial Animation using FDT

In this section we propose an approach compliant with the MPEG-4 standard to synthesize and control facial expressions generated using 3D facial models. This is achieved by establishing the MPEG-4 facial animation standard conformity with the quadratic deformation model representations of facial expressions. This conformity allows us to utilize the MPEG-4 facial animation parameters (FAPs) with the quadratic deformation tables, as a higher layer, to compute the FAP values. The FAP values for an expression E are computed by performing a linear mapping between a set of transformed MPEG-4 FAP points (using quadratic deformation models) and the 3D facial model semantics. The nature of the quadratic deformation model representations of

facial expressions can be employed to synthesize and control the six main expressions (surprise, fear, anger, sadness, disgust, and smile). Using Whissel's [109] psychological studies on emotions (described in Section 2.3.7) we compute an interpolation parameter that is used to synthesize intermediate facial expressions.

The rest of this section is organised as follows. Section 6.1.1 illustrates an overview of the MEPG-4 facial animation using quadratic deformation models. Section 6.1.2 explains the quadratic deformation model conformity with the MPEG-4 facial animation standard. Section 6.1.3 gives a description of how the MPEG-4 FAPs are computed using the quadratic models. Sections 6.1.4 and 6.1.5 describe a technique to generate different facial expressions.

6.1.1 Overview of MPEG-4 Facial Animation using FDT

In general, most MPEG-4 compliant facial models require a facial animation engine to derive the final rendering of the 3D facial appearance. The facial animation process starts by providing the facial animation engine with a 3D facial Model geometry and its semantic rules (such as FAPs). The engine then automatically computes the animation rules based on a stream of FAP values coming from a higher layer input. Finally, the 3D face is rendered based on the computed animation rules. The approach described in this section acts as the higher layer input that defines and computes the stream of FAP values based on the 3D model semantic rules and using the FDT parameters. The general picture of the process is shown in Figure 6.1.

The aim of our approach is to synthesize and control facial expressions using real muscle deformations of facial expressions (described in Chapter 3).

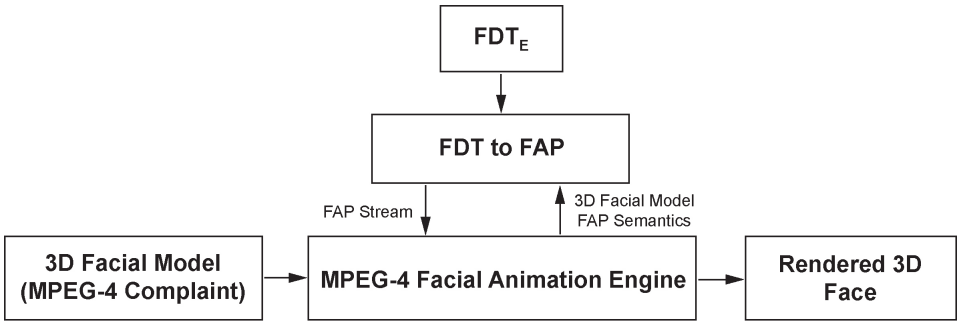


Figure 6.1: FDT to FAP mapping as part of the MPEG-4 facial animation engine.

To achieve this, we first check that the conformity of the quadratic models with the MPEG-4 facial animation standard. We then describe a mechanism to compute the mapping between the quadratic deformation models and FAPs. Finally, we describe an interpolation technique to synthesize and control facial expressions.

In the following sections we describe how the quadratic deformation model representations of facial expressions, can be used to synthesize and control the facial expression of a 3D facial model compliant with MPEG-4 facial animation standard.

6.1.2 The Quadratic Deformation Model Conformity with the MPEG-4 Facial Animation Standard

The derived FDT parameters are captured from the non-linear nature of the muscle deformations using quadratic deformation models. In contrast, the MPEG-4 standard was solely developed and standardised based on the FACS Action Units (AU). To be able to use the FDT parameters as a mechanism to control and synthesize facial expressions on the top of the MPEG-4 facial

animation standard we need to check for the conformity of the quadratic deformation models with the FACS AU description of muscle movements. In order to do this check the following process is carried out:

1. Generate a set of transformation constraint rules on the MPEG-4 FAPs that correspond to the AUs involved in the main universal expressions E (surprise, fear, anger, sadness, disgust, and smile). Figure 6.2 illustrates the constraint rules[1] for the transformations of the FAP points involved in each AU.

2. Map the defined muscle based facial regions, described in Section 3.1, to the corresponding FAP points as shown in Figure 6.3.

3. Apply the quadratic transformation of expression E (FDT_E) to the set of FAP points[2] .

4. Conformity is achieved if the transformed FAPs meets the constraint rules for each AU involved in E.

Using the process described above we have confirmed that the quadratic deformation models do conform to the AUs involved in each of the main universal facial expressions. This conformity allows us to utilize the MPEG-4 facial animation parameters with the quadratic deformation model representations (FDT) of facial expressions as illustrated in Figure 6.1.

The following section describes how the FAP values are computed from FDTs for different facial expressions.

[1] Constraint rules are based on the displacement of FAP_i and its transformation FAP_i'.

[2] The FAP points are normalised as described in Section 3.4.2.

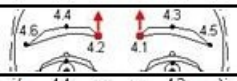
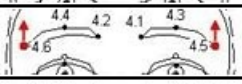
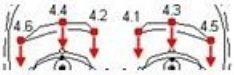
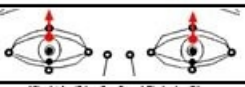
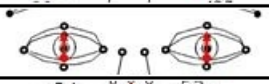
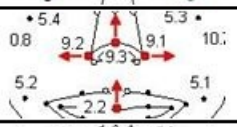
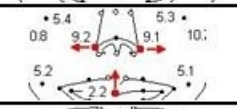
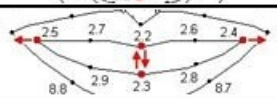
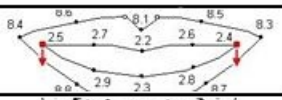
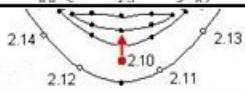
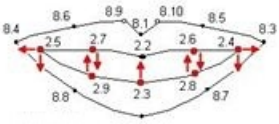
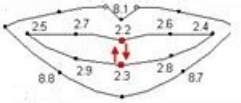
AU	AU Image	MPEG-4 FAP	Rules
1			[4.1y'-4.1y>0], [4.2y'-4.2y>0]
2			[4.5y'-4.5y>0], [4.6y'-4.6y>0]
4			[4.1y'-4.1y<0], [4.2y'-4.2y<0], [4.1x'-4.1x<0], [4.2x'-4.2x>0], [4.3y'-4.3y<0],[4.4y'-4.4y<0], [4.5y'-4.5y<0], [4.6y'-4.6y<0]
5			[3.2y'-3.2y>0], [3.1y'-3.1y>0]
6			[5.3y'-5.3y>0], [5.4y'-5.4y>0]
7			[3.3y'-3.3y>0], [3.4y'-3.4y>0]
9			[2.2y'-2.2y>0], [9.3y'-9.3y>0], [9.1x'-9.1.x>0], [9.2x'-9.2.x<0]
10			[2.2y'-2.2y>0], [9.1x'-9.1.x>0], [9.2x'-9.2.x<0]
12			[2.3y'-2.3y>0], [2.2y'-2.2.y<0], [2.4x'-2.4x>0], [2.5x'-2.5x<0]
15			[2.4y'-2.4y<0], [2.5y'-2.5y<0]
17			[2.10y'-2.10y>0]
20			[2.3y'-2.3y>0], [2.4x'-2.4x>0], [2.5x'-2.5x<0], [2.4y'-2.4y<0], [2.5y'-2.5y<0], [2.8y'-2.8y>0], [2.9y'-2.9y>0], [2.6y'2.6.y<0], [2.7y'-2.7y<0]
25			[2.2y'-2.2.y>0], [2.3y'-2.3.y<0]

Figure 6.2: Constraint rules for FAPs based on the AUs involved in the main universal expressions (AU images are adopted from [102][30], while the MPEG-4 FAP images are generated using the FAPs' illustration in [72]).

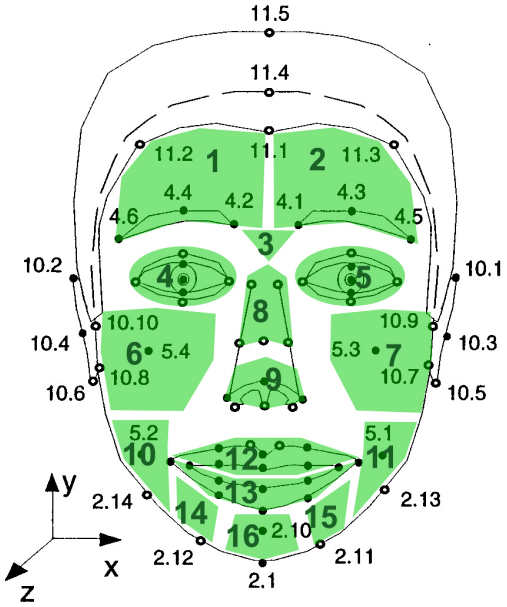


Figure 6.3: Facial muscle regions mapped to the corresponding FAP points.

6.1.3 FDT to FAP Mapping

In our approach, we map the defined muscle based facial regions into the corresponding FAP points[2] described by the MPEG-4 facial animation standard as shown in Figure 6.3. This allows us to generate different deformations and exaggerations of the FAP points to form the desired facial expression appearance.

For instance, if the FDT parameters of Expression E are applied to the regions shown in Figure 6.3, the position of the FAP points will transform to the desired expression E. However, to be able to deform a 3D facial model compliant with MPEG-4 facial animation standard, the displacement of each point has to be converted into a FAP value that matches the 3D facial model semantics. The following steps describe the process of converting the displacement of a facial point P_i to a FAP value defined by the semantics of a

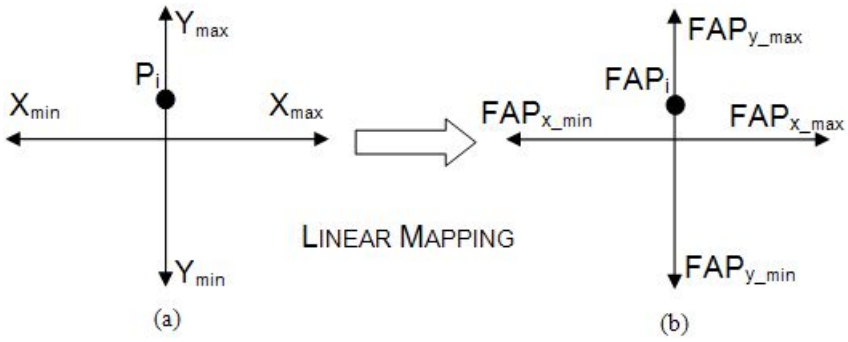


Figure 6.4: Mapping between (a) transformed facial points P_i using FDT and (b) its corresponding FAP_i point.

3D facial model.

- Deform a set of FAP facial points (which we represent as P_i) using the quadratic models as shown in Figure 6.3.

- Define the minimum and maximum displacement of the transformed facial points (P_i) in the xy-plane.

- Extract the corresponding minimum and maximum values of FAP_i from the 3D facial model semantics.

- Use linear mapping between the xy-plane values and the FAP-plane values to compute the FAP values. This process is described in Figure 6.4.

- The derived FAP values can then be used to synthesize facial expressions on the 3D facial model.

The pseudo-code below illustrates how the process shown in Figure 6.4 is computed mathematically using linear mapping with respect to the y-axis.

Listing 6.1

```
Integer getFAPValueY(y : Integer);
vars
 range, deformed_range : Integer;
 fap_range, fap_y_i : Integer;
 ratio : Decimal;
begin
 range = Ymax - Ymin;
 deformed_range = Pi - Ymin;
 ratio = deformed_range/range;
 fap_range = fap_y_max - fap_y_min;
 fap_y_i = (ratio*fap_range) + fap_y_min;
 return fap_y_i;
end;
```

Figure 6.6 shows an example of mapping the FDT parameters of several facial expressions to Greta's facial model. The ability to map the FDT parameters of main facial expressions allows us to control the expressivity of the facial features and also generate intermediate facial expressions. The following sections describe in more detail how the expressivity of the facial features is controlled.

6.1.4 Expressivity of the Facial Appearance

The way the FDTs are defined allows us to control the expressiveness of the 3D facial model's appearance and its level of exaggeration. Similar to the definition described in Section 5.3.1, the interpolation of the FDT parameters values between the neutral expression and desired expression allows for the generation of different expressivity levels. An exaggerated facial expression can also be generated by extrapolating beyond the desired expression FDT parameters. We re-use the parametric linear interpolation Equations 5.3 and 5.4, defined in Section 5.3.1, between a neutral expression N, and a given expression E.

$$\begin{aligned} x'_i &= (1-t)x'_{i(N)} + tx'_{i(E)} \\ &= \sum_{u=0}^{2}\sum_{v=0}^{2}((1-t)A_{uv(N)} + tA_{uv(E)})x_i^u y_i^v \end{aligned} \tag{6.1}$$

$$\begin{aligned} y'_i &= (1-t)y'_{i(N)} + ty'_{i(E)} \\ &= \sum_{u=0}^{2}\sum_{v=0}^{2}((1-t)B_{uv(N)} + tB_{uv(E)})x_i^u y_i^v \\ & \qquad\qquad 0 \leq t \leq 1 \end{aligned} \tag{6.2}$$

The parameter t defines the degree of the facial expression E and it is used for producing different facial expressions on the 3D facial model. The value of t could be increased above 1 to extrapolate the coefficients beyond the normal values for that expression. Our approach allows the user to interpolate the parameter t by varying the value between $0 \leq t \leq 1$ or extrapolate the value $t > 1$. A continuous variation of the parameter t

produces an animation sequence for the given facial expression. An example of the animation sequence for the facial expression 'surprise' is shown in Figure 6.8.

6.1.5 FDT for Intermediate Expressions

The ability of using Equations 6.1 and 6.2 to interpolate between the facial deformation parameters for an expression E, allows us to generate different intensity levels of the facial expression E. These expressions can be interpreted as intermediate expressions of E or expressions that belong to the same category of E.

Whissel [109] defined intermediate expressions I of a facial expression E on a two dimensional axis, labelled Activation and Evaluation, as shown in Figure 6.5. Table 6.1 shows an example of some of the activation and evaluation values represented by Whissel's study. The evaluation levels of a facial expression express the internal feelings of an individual, which can be difficult to define using facial muscles. However, the activation level can be directly associated with facial muscles and their movements.

Table 6.1: Example of emotion activation and evaluation values from Whissel's study [109].

Emotion	Activation	Evaluation
Angry	4.2	2.7
Hostile	4	1.7
Furious	5.6	3.7

Using the activation levels (defined by Whissel) we can generate the intermediate facial expressions I of a given expression E by mapping their

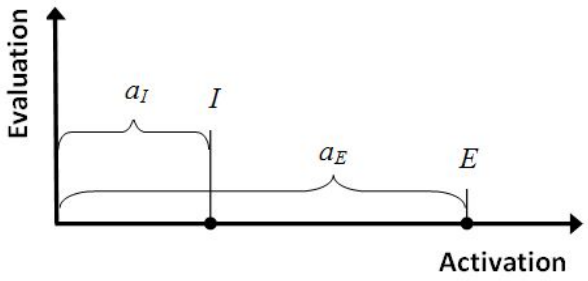


Figure 6.5: Whissel's definition of facial expressions in two dimensional space (Activation and Evaluation).

activation level to the FDT parameters. This can be achieved by computing the parameter t for Equations 6.1 and 6.2 as follows:

$$t = (a_I / a_E) \tag{6.3}$$

Where the a_I and a_E are the activation levels for the intermediate facial expression I and the main facial expression E respectively.

6.2 Results using the Greta ECA Facial Model

In this section we illustrate the results of utilizing the Greta ECA engine with our approach. Figure 6.6 shows an example of Greta's face expressing all of the main universal facial expressions (surprise, fear, anger, sadness, disgust, and smile). Figure 6.7 gives an example of Greta showing different intermediate expressions generated using the activation parameters defined by Whissel's study [109] as shown in Table 6.1. Figure 6.8 demonstrates an animation of the surprise expression achieved by interpolating the FDT parameters between the neutral and surprise expressions.

Table 6.2: Confusion matrix for the primary facial expression shown by the Greta ECA.

Expression	**Smile**	**Anger**	**Sad**	**Surprise**	**Fear**	**Disgust**
Smile	**100.0%**	0.0%	0.0%	0.0%	0.0%	0.0
Anger	5.6%	**72.2%**	11.1%	0.0%	0.0%	11.1%
Sad	0.0%	5.6%	**88.8%**	0.0%	0.0%	5.6%
Surprise	0.0%	0.0%	0.0%	**72.2%**	27.8%	0.0%
Fear	0.0%	0.0%	5.6%	38.8%	**55.6%**	0.0%
Disgust	5.6%	5.6%	0.0%	5.6%	22.2%	**61.0%**

The primary facial expression mapped on Greta's face using the quadratic deformation models are evaluated using a subjective study. We asked 18 subjects to look at Greta's primary facial expressions one by one and to choose a label of from the following for each expression: (surprise, fear, anger, sadness, disgust, and smile). The results of the participants are summarised in Table 6.2. The results show that Greta's expressions were correctly classified with an accuracy of 71.75%. The accuracy results show a highest classification of 100% for smile and a lowest of 55.6% for fear. The evaluation results coincide closely with the results of Table 4.2 (from Section 4.3.1) of the data images used to derive the quadratic deformation model parameters.

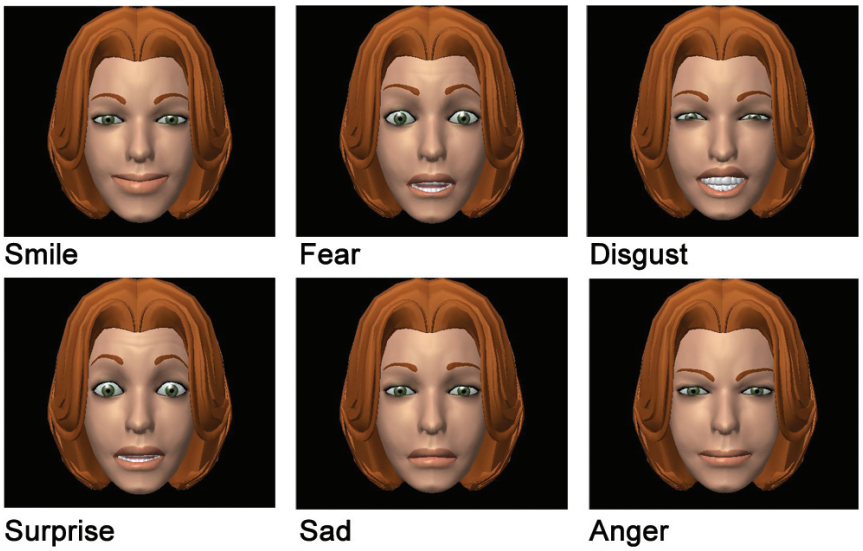


Figure 6.6: Greta expressing the six main universal facial expressions described by Paul Ekman [29].

Figure 6.7: Intermediate expressions of the Anger Facial Expression.

Figure 6.8: A facial animation sequence of the surprise facial expression.

Chapter VII

Conclusion and Future Work

This thesis proposed a novel approach to represent facial expressions based on a quadratic deformation model applied to muscle regions. The rubber-sheet transformation functions are used to compute the deformation parameters for facial regions, where we defined sixteen facial regions based on the anatomy of the facial muscle system. The results form a set of facial deformation coefficient tables (FDT), which we generalized by taking the average of the FDTs for each expression. The FDTs can be used in facial expression applications such as facial recognition and facial animations.

A novel facial expression recognition method was developed by utilizing an active appearance model based facial feature tracking system with quadratic deformation model representations of facial expressions. The method tracks 37 facial feature points corresponding to the layout of the MPEG-4 standard FAPs. A reference of the neutral face facial feature points are extracted and deformed according to the FDTs. Applying the Euclidean distance similarity measures on the tracked points and the reference deformations allows us to recognize facial expression. Our results show that the overall average facial expression recognition rate is 88.9%. The achieved accuracy clearly shows encouraging results and can lead to future direction for higher rates in facial expression analysis and recognition. We evaluated the

data acquired to derive the quadratic deformation models and found that the accuracy of the fear expression may have been affected by the fact that the acted expressions were not preformed correctly.

Furthermore, we proposed an approach that mainly contributes to the field of caricature generation and rendering. Its novelty comes from being able to manipulate the facial appearance and expressivity of the caricature using the elasticity of real facial muscle deformation. In addition, our approach employs geometric moment functions in a stroke based rendering algorithm, which enables the creation of artistically rendered expressive facial caricatures and does not require any earlier training of caricature prototypes drawn by artists.

To demonstrate the use of the quadratic deformation model to animate 3D faces, we proposed an approach to synthesize and control 3D facial models, compliant with the MPEG-4 facial animation, using quadratic deformation model representations of facial expressions. We conform the quadratic deformation model with the FACS AU for the purpose of utilizing the MPEG-4 facial animation standard with the quadratic representations. Facial expressions are synthesized by computing the FAP values based on the semantics of the 3D facial model. The FAP values for a facial expression are computed by performing linear mapping between a deformed set of FAP points (using FDT) and the 3D facial model semantics. We use the activation level of Whissel's study [109] along with our quadratic deformation model interpolation technique to generate not only the main universal expressions but also intermediate expressions. The achieved results are encouraging and can lead to future work in the field of expressive 3D facial models.

To evaluate and show the success of the proposed quadratic model representations, a qualitative comparative analysis of the quadratic deformation models and the commonly used deformation models was carried out. Section 7.1 describe the details of the comparative analysis.

7.1 Quadratic Deformation Model Comparative Analysis

To evaluate and show the effectiveness of the proposed quadratic deformation model representations, a qualitative comparison analysis is performed to compare the following properties and characteristics [54]:

- Computational effort: The efficiency of the algorithm to generate a deformation.
- Model complexity: The mathematical complexity of the model.
- Geometry invariance: The model's flexibility when the face's geometry changes. This can also be described as how generic the model is when mapped to different face models.
- Physical realism: Refers to how realistic the physical functionality and mechanism of the model is.
- Visual Realism: Is a subjective measure of the visual results achieved by the model?
- User Control: Describes the level of difficulty of how easy to generate results using the model.

- Control Parameterization: The method used to control the visual appearance when using the model.

Table 7.1 shows the comparative analysis of the commonly used facial deformation models and the proposed quadratic deformation models. The comparative results show that the main advantages of the proposed quadratic deformation model is in achieving a medium to high physical and visual realism with low computational and model complexity. Moreover, the proposed models are generic and are geometrically independent and are easy to control by the user to achieve the desired deformation.

The results from Sections 5.5, 6.2 and Table 6.2 give examples of how the deformation models can map expressions on to face sketches and face models with high visual realism. Finally, the generality of the quadratic deformation models allows it to be used in different facial application areas. This includes animation, synthesis, electronic entertainment or recognition systems.

Table 7.1: Comparative analysis of the quadratic deformation model and the commonly used deformation models [54].

Model	Computational Effort	Model Complexity	Geometry Invariance	Physical Realism	Visual realism	User Control	Control Parameterization
Interpolation	Low	Low	High	Low	Med to High	Easy	Interpolation
Parametric	Low	Low	High	Low	Med	Easy	Interpolation
Physics-based models							
- Spring	Med	Med	Low - Med	Med	Low - Med	Med	FACS
- Vector	Med	Med	Low - Med	Med	Med	Med - High	FACS
- Layered	High	High	Low - Med	High	Med - High	Hard	FACS
Pseudo-based models							
- Procedures	Low	Low	High	Med	Med	Easy	-
- Spline	Low - Med	Low	Low	Med	Med	Easy	FACS
- Free Form Deformation	Low - Med	Low	Low	Med	Med	Easy	MPA
- **FDT deformation**	**Low**	**Low**	**Low**	**Med-High**	**Med-High**	**Easy**	**FDT**
FEM	High	High	Med	High	Med - High	Hard	FACS

7.2 Limitations and Future Work

In this section we describe some of the limitations of the proposed quadratic deformation models and the different applications developed.

7.2.1 Data Acquisition

The derived quadratic deformation models parameters rely on the acquired data from subjects acting the primary facial expressions. We found that the fear expression may have been poorly acted by subjects which largely influenced the evaluation results of the different systems developed. Future work would be to gather data from subjects, performing the primary facial expressions, who are expert actors. This will improve the quality of the facial expression representations.

At the moment, the facial expression data used to derive the quadratic deformation models are acquired based on photographs. An interesting future research direction would be to collect the data from actors performing facial expressions using motion capture (mocap) systems. Using mocap data to analyze and derive the quadratic deformation models will add a third dimension to the quadratic facial expression models, which can result in a more powerful set of parameters to represent facial expressions.

7.2.2 Facial Deformation Tables

The facial deformation tables (FDT) are a set of 12×16 parameter arrays. They represent the deformation of 16 regions using the 12 coefficients of the quadratic equations; however, according to the FACS description of facial expressions [29, 31], the primary facial expressions can be represented by

moving certain regions of the face. This means that only those regions, that are influenced, need to be contained in the FDT. Moreover, the 16 regions may not all need to be represented using quadratic deformation equations, as some regions will only deform in a linear fashion and can be represented by linear equations. Future research in the direction of studying how to compress the size of the FDT representations can minimize the number of parameters used.

7.2.3 Facial Expression Recognition

A future direction to this research is to eliminate the manual process of registering the reference natural expression to fully automate the facial recognition process. One could also research the minimum number of tracked feature points required to maintain high recognition rates. Finally, further work can be done on improving the accuracy performance by identifying distinct deformations of facial features for different expressions.

7.2.4 Expressive Caricatures

A limitation of the caricature generation system is the manual process involved in identifying the hair and the ears using the digital matting algorithm of Sindeyev and Konushin [94]. Future work will focus on eliminating this process by developing an automatic hair and ears segmentation algorithm.

Future directions for this research are also directed towards conducting subjective evaluations of the generated caricatures to study their appearance, exaggeration, and the resemblance of the facial expressions. We believe that the proposed method can also be further extended to produce pencil sketch

rendering of images and cartoon-like 3D characters.

7.2.5 Animating 3D Faces

We used the facial expression study described by Whissel [109] to generate intermediate facial expressions. However, the proposed approach is limited to generating different intermediate expressions, using FDT parameters, based on one primary expression. This condition limits the representation of other facial expressions that belong to two emotional expression categories. Future research direction can focus on further enriching the type of expressions that can be generated using the quadratic deformation models. One possibility is to look at the using the emotion wheel described by Plutchik [81] to generate facial expressions belonging to two different emotions.

Evaluations were conducted using static expressions imposed on Greta's face, however, it will be interesting to investigate if better results can be achieved by evaluating animated sequences of Greta's face.

Appendix A

Facial Feature Tracking

The Active Appearance Models (AAM) facial feature tracking system used in this thesis is the work of Dr. Jason Saragih and Dr. Roland Goecke [88]. The content of this chapter is presented by Dr. Roland Goecke as part of our collaboration with him.

The AAM's intrinsic variations in shape and texture of deformable visual objects are modelled as a linear combination of basis modes of variation that are composed with a global (similarity) transformation:

$$S(q_s) : \Re^{n_s} \mapsto \Re^{2n} = s(I \otimes R)(\mu_s + \phi_s p_s) + \mathbf{1} \otimes t) \quad \text{(A.1)}$$

$$T(q_s) : \Re^{n_t} \mapsto \Re^{m} = \alpha(\mu_t + \phi_t p_t) + \beta\mathbf{1} \quad \text{(A.2)}$$

where S and T denote the generative models for shape and texture, parameterized by $q_s = \{s, \mathbf{R}, \mathbf{t}, \mathbf{p_s}\}$ and $q_t = \{\alpha, \beta, p_t\}$, respectively. Here, $\{\mu_s, \phi_s\}$ and $\{\mu_t, \phi_t\}$ denote the mean and bases of variations of the shape and texture, which are typically obtained by applying Principal Component Analysis (PCA) on the training data. The intrinsic shape is composed with a similarity transform, parameterized by a global scaling s, a rotation $\mathbf{R}$ and a translation $\mathbf{t}$. The intrinsic texture is scaled by a global gain α and biased by β. Finally, n denotes the number of points in the model's shape and m

denotes the number of pixels in the model's texture.

AAM fitting (or tracking for image sequences) is the process of finding the model parameters $p = \{q_s, q_t\}$ which best fit an AAM to an image I. This is usually an iterative procedure that sequentially updates the model parameters $\mathbf{p}$ through an update function

$$\Delta p = U(*, p) \circ F(I; p) \tag{A.3}$$

Here, F is a feature extraction function that represents the image I from the perspective of the AAM at its current parameter settings and Δp are the updates to be applied to the current parameters. U is typically chosen to be the linear update model,

$$U(F; p) : \Re^m \mapsto \Re^{n_s+n_t} = Gf + b \tag{A.4}$$

where $f = F(I; p)$, although nonlinear mappings have also been used [89]. In any case, a good coupling between U and F is required to ensure good predictions of the updates.

Recently, Saragih and Goecke [88, 87] proposed learning the entire fitting procedure in a discriminative framework rather than extracting the update model. Learning is performed on examples of real fitting scenarios, simulated on the training data. Utilizing a different update model in each iteration, the parameter updates can be written as

$$\Delta p_i = G_i F\Big(I; p + \sum_{j=1}^{i-1} \Delta p_j\Big) + b_i \tag{A.5}$$

where $\{G_i, b_i\}$ is the fixed update model for the i-th iteration. Given the training set, the optimal update models for all k iterations can be found by minimizing a cost function over the hand-labelled annotations for the j-th training sample,

$$p^* = min\ q_s \|s^* - S(q_s)\|^2 \tag{A.6}$$

The parameters of this cost function are the update models themselves. A distance function that penalizes the difference between the manually annotated shapes and the predicted model's shape after N_i iterations is used [87]. Compared to texture based error measures, commonly used in generative AAM fitting, this distance function better encompasses all available knowledge about the optimal parameter setting, i.e. the hand-labeled annotations. With this formulation, the training procedure essentially simulates real fitting problems on the set of training images and perturbations.

Having trained the update models, AAM fitting then proceeds by simply applying all pre-computed update models for all trained iterations with no early termination. If the unseen images and their perturbations resemble those in the training set, then the fitting performance of the minimizer can be expected to approach that at training. This discriminative-iterative approach boasts significant improvements over other AAM fitting methods in both convergence accuracy and rate. Furthermore, these improvements are afforded without sacrificing computational efficiency. Also, the method affords excellent generalizability, as evidenced by its high convergence rates [87].

Appendix B

Geometric Moments

The content of this chapter explains the theory of geometric moments and is based on my masters thesis work [69].

The first major work on image moments was published by Hu [47] in 1962. He presented a theory of two-dimensional geometric moment invariants which were used for alphabetical character recognition. Thereafter, significant contributions to the application of moment invariants have appeared. For example, moment invariants have been used in aircraft identification [26], ship identification [95], pattern recognition [3], robotic motion [40], and character recognition [14, 112, 113]. The geometric moment invariants derived by Hu have also been extended to larger sets by Wong and Siu [113] and to other moment invariant types (Teague [97], Abu-mostafa [1], Reddi [85]).

Geometric moments M_{pq} of order $(p+q)$, of an image intensity function $f(x,y)$, can generally be defined as follows:

$$M_{pq} = \int\int_{\zeta} x^p y^q f(x,y) \quad dx\, dy, \qquad\qquad p,q = 0,1,2,3,\ldots \tag{B.1}$$

where ζ denotes the pixels of an image region in the xy-plane in which the function $f(x,y)$ is defined. The monomial product $x^p y^q$ is the basis function for the moments definition in ζ.

In the case of a digital image, the double integral in equation B.1 is replaced by a summation. The geometric moments for an $(N \times N)$ image is given by:

$$M_{pq} = \sum_{x}^{N} \sum_{y}^{N} x^p y^q I(x, y) \tag{B.2}$$

where N is the size of the image and $I(x, y)$ is the image intensity function.

Shape Features using Moments

Different orders of geometric moments describe different shape features of an image intensity distribution. The image moment of the p_{th} degree about the $x-$axis and the q_{th} degree about the $y-$axis is defined in equation B.2. The following is a list of some image moments and how they describe image features:

- The moment of the zeroth order (M_{00}) represents the total intensity of the image intensity distribution. If the image contains an object, M_{00} represents the area of the object.

- First order moments M_{10}, M_{01} represent the intensity moments of an image about its $y-$axis, and $x-$axis respectively. Moments M_{10} and M_{01} combined with M_{00} represent the intensity centroid (x_c, y_c) as shown in equation B.3.

- The second order moments M_{20}, M_{02}, and M_{11} represent the orientation of the image as shown in equation B.4.

The principal shape features (Figure B.1) are computed as follows [67, 40]:

$$x_c = \frac{M_{10}}{M_{00}};\ y_c = \frac{M_{01}}{M_{00}} \tag{B.3}$$

$$\theta = \frac{\tan^{-1}(\frac{b}{a-c})}{2} \tag{B.4}$$

$$w = \sqrt{6(a + c - \Delta)} \tag{B.5}$$

$$l = \sqrt{6(a + c + \Delta)} \tag{B.6}$$

where a, b, c and Δ are defined as,

$$a = \frac{M_{20}}{M_{00}} - x_c^2 \tag{B.7}$$

$$b = 2(\frac{M_{11}}{M_{00}} - x_c y_c) \tag{B.8}$$

$$c = \frac{M_{20}}{M_{00}} - y_c^2 \tag{B.9}$$

$$\Delta = \sqrt{b^2 + (a - c)^2} \tag{B.10}$$

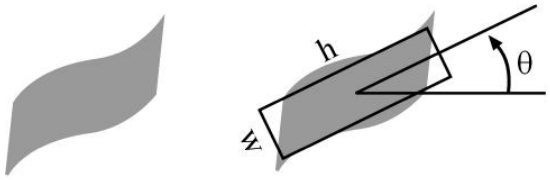


Figure B.1: A shape and its equivalent rectangle.

Appendix C

Statement of Contributions in Published Work

Contributions described in this thesis are fully my own, with the exception of collaborated work with other researchers. The following clarifies collaborated work and my contributions in the published work.

- **Chapter 3 - Facial Expression Representation Using Quadratic Deformation Models:** The work on the main contribution within this thesis was entirely my own and it was assisted by my supervisor, Dr. Mukundan.

  I developed the software to derive and validate the quadratic deformations. I received assistance from Karin Fitz[1] to help me with taking photographs of subjects and collecting the facial expression data. I wrote the majority of the paper, published at CGIV 2009, with the assistance of my supervisory team.

  The published paper at IVCNZ 2009 was part of a final year computer-engineering project conducted by ChiangHau Tay[2]. I assisted ChiangHau throughout his project to generate facial expressions on face portraits.

[1] Intern at the HITLab NZ.

[2] Electrical and Computer Engineering Department, University of Canterbury, Christchurch, New Zealand.

- **Chapter 4 - Facial Expression Recognition using Quadratic Deformation Models:** The described new approach for classifying facial expressions by using a facial feature tracking system was my idea and it was assisted by my supervisor, Dr. Mukundan. I developed the software implementation of the system, with the exception of the Active Appearance Model (AAM) facial feature tracking system. The AAM tracking system belongs to Dr. Jason Saragih[3] and Dr. Roland Goecke[4] [5]. I collaborated mainly with Dr. Roland Goecke who provided me with the software libraries and the training on how to use the system.

  The research work was partly funded through the FRST Grant UOCX0608 and was part of the CALLAS project that myself and Dr. Hartmut Seichter [6] were involved in.

  The published paper at DICTA 2009 was written with the assistance of my supervisors and Dr. Hartmut Seichter. The facial feature tracking section of the paper was presented by Dr. Roland Goecke.

- **Chapter 5 - Expressive Caricatures using Quadratic Deformation Models:** The presented approach for generating expressive caricatures was entirely my idea and it was assisted by my supervisor, Dr. Mukundan. I developed the software implementation of the expressive generation system. The stroke-based painterly rendering algorithm

[3] Carnegie Mellon University, Pittsburgh, USA.

[4] HCC Lab / NCBS, Faculty of Information Sciences and Engineering, University of Canberra, Australia.

[5] School of Computer Science, CECS, Australian National University, Canberra, Australia.

[6] HITLab New Zealand, University of Canterbury, Christchurch, New Zealand.

is based on my masters thesis work, with some enhancements that I made to allow for the generation of rendered caricatures. I wrote the majority of the published work at SIGGRAPH 2010 and IEEE ICCSIT 2010 with the assistance of my supervisors.

I recieved assistance at the initial stages of this work from Daniel Lond [7] to work on the background edge detection using the OpenCV library. The published paper at ACE 2009 was written by myself and Daniel Lond and was assisted by my supervisors.

- **Chapter 6 - Facial Animation using Quadratic Deformation Models:** Using the mathematical functions to synthesize facial expressions on 3D faces was my own work. I collaborated with Prof. Catherine Pelachaud[8] and her team who provided me with an embodied conversational agent (Greta) that is compliant with the MPEG-4 facial animation standards, for experimental analysis.

  The research work was partly funded through the FRST Grant UOCX0608 and is part of the EU FP6 Integrated Project CALLAS IP-CALLAS IST-034800.

  I wrote the majority of the published paper, at CGIV 2010, with the assistance of my supervisors and Prof. Catherine Pelachaud.

[7] Department of Mathematics and Statistics, University of Canterbury, New Zealand.

[8] CNRS-LTCI, Telecom ParisTech, Paris, France.

References

[1] Y.S. Abu-Mostafa and D. Psaltis. Recognitive aspects of moment invariants. In *IEEE Transactions Pattern Analysis and Machine Intelligence*, volume 11, pages 698–706, 1984.

[2] E. Akleman. Making caricatures with morphing. In *SIGGRAPH '97: ACM SIGGRAPH 97 Visual Proceedings: The art and Interdisciplinary Programs of SIGGRAPH '97*, page 145, New York, NY, USA, 1997. ACM.

[3] F.L. Alt. Digital pattern recognition by moments. In *JACM*, pages 240–258, 1962.

[4] K. Arai, T. Kurihara, and K. Anjyo. Bilinear interpolation for facial expression and metamorphosis in real-time animation. *The Visual Computer*, 12:105–116, 1996. 10.1007/BF01725099.

[5] M. Argyle. *Bodily communication*. Methuen, 1988.

[6] K. Balc. Xface: Open source toolkit for creating 3d faces of an embodied conversational agent. In *Smart Graphics*, pages 263–266, 2005.

[7] A.H. Barr. Global and local deformations of solid primitives. *SIGGRAPH Computer Graphics*, 18(3):21–30, 1984.

[8] J.N. Bassili. Emotion recognition: The role of facial movement and the relative importance of upper and lower areas of the face. *Journal of Personality and Social Psychology*, 37(11):2049 – 2058, 1979.

[9] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *SIGGRAPH '99: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, pages 187–194, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[10] F. Bourel, C.C. Chibelushi, and A.A. Low. Robust facial feature tracking. In *Proceedings of the 11th British Machine Vision conference BMVC2000*, volume 1, pages 232–241, Bristol, England, September 2000.

[11] S. Brennan. *Caricature Generator*. Masters thesis, Cambridge, MIT, 1982.

[12] A.J. Calder, A.M. Burton, P. Miller, A.W. Young, and S. Akamatsu. A principal component analysis of facial expressions. *Vision Research*, 41(9):1179–1208, April 2001.

[13] J. Canny. A computational approach to edge detection. *IEEE Transactions Pattern Analysis Machine Intelligence*, 8(6):679–698, 1986.

[14] G.L. Cash and M. Hatamian. Optical character recognition by the method of moments. *Computer Vision, Graphics, and Image Processing*, 39(3):291–310, 1987.

[15] J. Chai, J. Xiao, and J. Hodgins. Vision-based control of 3d facial animation. In *SCA '03: Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 193–206, Aire-la-Ville, Switzerland, 2003. Eurographics Association.

[16] H. Chen, Z. Liu, C. Rose, Y. Xu, H. Shum, and D. Salesin. Example-based composite sketching of human portraits. In *Proceedings of the 3rd International Symposium on Non-Photorealistic Animation and Rendering*, NPAR '04, pages 95–153, New York, NY, USA, 2004. ACM.

[17] H. Chen, Y. Xu, H. Shum, S. Zhu, and N. Zheng. Example-based facial sketch generation with non-parametric sampling. *IEEE International Conference on Computer Vision*, 2:433, 2001.

[18] H. Chen, N. Zheng, L. Liang, Y. Li, Y. Xu, and H. Shum. Pictoon: a personalized image-based cartoon system. In *MULTIMEDIA '02: Proceedings of the Tenth ACM International Conference on Multimedia*, pages 171–178, NY, USA, 2002. ACM.

[19] P. Chiang, W. Liao, and T. Li. Automatic caricature generation by analyzing facial features. *Asian Conference on Computer Vision*, 2004.

[20] T. Cootes. "Software, Modeling and Search Software". Retrieved December, 2008, from `http://personalpages.manchester.ac.uk/staff/timothy.f.cootes`.

[21] M.N. Dailey and G.W. Cottrell. PCA = Gabor for expression recogni-

tion. Technical Report 0629, University of California at San Diego, La Jolla, CA, USA, 1999.

[22] DataFace. "Facial Muscles". Retrieved March, 2009, from `http://www.face-and-emotion.com/dataface/expression/muscles.jsp`.

[23] Z. Deng, P. Chiang, P. Fox, and U. Neumann. Animating blendshape faces by cross-mapping motion capture data. In *I3D '06: Proceedings of the 2006 symposium on Interactive 3D Graphics and Games*, pages 43–48, New York, NY, USA, 2006. ACM.

[24] Z. Deng and J. Noh. Computer facial animation: A survey. In Z. Deng and U. Neumann, editors, *Data-Driven 3D Facial Animation*, pages 1–28. Springer London, 2007.

[25] F. Dornaika and J. Ahlberg. Efficient active appearance model for real-time head and facial feature tracking. In *AMFG '03: Proceedings of the IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, page 173, Nice, France, 2003. IEEE Computer Society.

[26] S. Dudani, K. Breeding, and R. McGhee. Aircraft identification by moment invariants. In *IEEE Transactions on Computers*, pages vol. C–26, 39–46, 1977.

[27] G.J. Edwards, C.J. Taylor, and T.F. Cootes. Interpreting face images using active appearance models. In *FG '98: Proceedings of the 3rd International Conference on Face & Gesture Recognition*, pages 300–305, Nara, Japan, April 1998. IEEE.

[28] P. Eisert and B. Girod. Analyzing facial expressions for virtual conferencing. *IEEE Computer Graphics and Applications*, 18:70–78, 1998.

[29] P. Ekman. *Unmasking the face.* Prentice Hall., 1975.

[30] P. Ekman and W.V. Friesen. Facial Action Coding System (FACS): Manual. *Palo Alto: Consulting Psychologists Press*, 1978.

[31] P. Ekman, W.V. Friesen, and J. C. Hager. Facial Action Coding System (FACS): 2nd Ed. *Salt Lake City, UT: Research Nexus EBook*, 2002.

[32] I. Essa. "Thesis: Irfan Essa's PhD Thesis (1994): Analysis, interpretation and synthesis of facial expressions". Retrieved August, 2010, from `http://prof.irfanessa.com/`.

[33] I.A. Essa. *Analysis, interpretation and synthesis of facial expressions.* PhD thesis, Massachusetts Institute of Technology. Department of Architecture. Program in Media Arts and Sciences, Cambridge, MA, USA, 1995.

[34] I.A. Essa and A.P. Pentland. Coding, analysis, interpretation, and recognition of facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:757–763, 1997.

[35] G. Faigan. *The Artist's guide to Facial Expressions.* Watson-Guphill Publications, 1990.

[36] B. Fasel and J. Luettin. Automatic facial expression analysis: a survey. *Pattern Recognition*, 36:259–275, 1999.

[37] Alan Fielding. "Distance and similarity measures". Retrieved June, 2011, from `http://alanfielding.co.uk/multivar/dist.htm`.

[38] V.C. Flores. ARTNATOMY (Anatomical basis of facial expression interactive learning tool). In *SIGGRAPH '06: ACM SIGGRAPH 2006 Educators program*, page 22, Boston, Massachusetts, 2006. ACM. doi: 10.1145/1179295.1179318.

[39] D.R. Forsey and C.L. Wang. Langwidere: A new facial animation system. Technical report, University of British Columbia, Vancouver, BC, Canada, Canada, 1994.

[40] W.T. Freeman, D.B. Anderson, P.A. Beardsley, C.N. Dodge, M. Roth, C.D. Weissman, W.S. Yerazunis, H. Kage, K. Kyuma, Y. Miyake, and K. Tanaka. Computer vision for interactive computer graphics. *IEEE Computer Graphics and Applications*, 18(3):42–53, 1998.

[41] T. Fujiwara, M. Tominaga, K. Murakami, and H. Koshimizu. Web-picasso: Internet implementation of facial caricature system picasso. In *ICMI '00: Proceedings of the Third International Conference on Advances in Multimodal Interfaces*, pages 151–159, London, UK, 2000. Springer-Verlag.

[42] R.C. Gonzalez and R.E. Woods. *Digital Image Processing*. Prentice-Hall, 3rd edition, 2007.

[43] B. Gooch, E. Reinhard, and A. Gooch. Human facial illustrations: Cre-

ation and psychophysical evaluation. *ACM Transactions on Graphics*, 23(1):27–44, 2004.

[44] A. Hertzmann. Algorithms for rendering in artistic styles. PhD thesis, New York University, 2001.

[45] IvyRose Holistic. "Facial Muscles". Retrieved July, 2010, from `http://www.ivy-rose.co.uk/HumanBody/Muscles/FacialMuscles.php`.

[46] P. Hong, Z. Wen, and T. Huang. Speech driven face animation. In *MPEG-4 facial animation - the standard, implementations, and applications*. John Wiley & Sons, Inc, 2002.

[47] M. Hu. Visual pattern recognition by moment invariants. In *IRE Transactions on Information Theory*, pages 8:179–187, 1962.

[48] A. Hughes and A. Gair. *Caricatures: Everything You Need to Know to Get Started*. HarperCollins Publishers, London, 1999.

[49] S. Iwashita, Y. Takeda, and T. Onisawa. Expressive facial caricature drawing. *IEEE International Conference on Fuzzy Systems*, 3, 1999.

[50] J.Ghent and J. McDonald. Photo-realistic facial expression synthesis. *Image and Vision Computing*, 23(12):1041 – 1050, 2005.

[51] P. Joshi, W.C. Tien, M. Desbrun, and F. Pighin. Learning controls for blend shape based realistic facial animation. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Courses*, page 17, New York, NY, USA, 2006. ACM.

[52] K. Kähler. *3D Facial Animation- Recreating human heads with virtual skin, bones, and muscles*. VDM Verlag, Germany, 2007.

[53] K. Kähler, J. Haber, and H. Seidel. Geometry-based muscle modeling for facial animation. In *GRIN'01: No description on Graphics interface 2001*, pages 37–46, Toronto, Ont., Canada, Canada, 2001. Canadian Information Processing Society.

[54] P. Kalra, S. Garchery, and S. Kshirsagar. Facial deformation models. In *Handbook of Virtual Humans Edited by N.M. Thalmann and D. Thalmann*, pages 119–139. John Wiley & Sons, 2004.

[55] P. Kalra, A. Mangili, N.M. Thalmann, and D. Thalmann. Simulation of facial muscle actions based on rational free form deformations. In *Computer Graphics Forum*, volume 11 of *Computer Graphics Forum (Netherlands)*, pages 59–69, 1992. MIRALab., Geneva Univ., Switzerland.

[56] T. Kanade, Y. Yingli, and J.F. Cohn. Comprehensive database for facial expression analysis. In *FG '00: Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition 2000*, page 46, Washington, DC, USA, 2000. IEEE Computer Society.

[57] T. Kanade, Y. Yingli, and J.F. Cohn. Comprehensive database for facial expression analysis. In *FG '00: Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition 2000*, page 46, Washington, DC, USA, 2000. IEEE Computer Society.

[58] T.B. Kuederle. Muscle-Based Facial Animation. Masters Thesis, University of Applied Science Wedel, Germany, 2005.

[59] Y. Lee, D. Terzopoulos, and K. Waters. Realistic modeling for facial animation. In *SIGGRAPH '95: Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, pages 55–62, NY, USA, 1995. ACM.

[60] J. P. Lewis, J. Mooser, Z. Deng, and U. Neumann. Reducing blendshape interference by selected motion attenuation. In *Proceedings of ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (I3DG*, pages 25–29, 2005.

[61] S.Z. Li, A.K. Jain, Y. Tian, T. Kanade, and J.F. Cohn. Facial expression analysis. In *Handbook of Face Recognition*, pages 247–275. Springer New York, 2005.

[62] W. Liao and C. Lai. Automatic generation of caricatures with multiple expressions using transformative approach. In Ozgur Akan, Paolo Bellavista, Jiannong Cao, Falko Dressler, Domenico Ferrari, Mario Gerla, Hisashi Kobayashi, Sergio Palazzo, Sartaj Sahni, Xuemin (Sherman) Shen, Mircea Stan, Jia Xiaohua, Albert Zomaya, Geoffrey Coulson, Fay Huang, and Reen-Cheng Wang, editors, *Arts and Technology*, volume 30 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 263–271. Springer Berlin Heidelberg, 2010.

[63] J.C. Lucero and K.G. Munhall. A model of facial biomechanics for

speech production. *The Journal of the Acoustical Society of America*, 106(5):2834–2842, 1999.

[64] L .Malatesta, A. Raouzaiou, K. Karpouzis, and S. Kollias. Mpeg-4 facial expression synthesis. *Personal Ubiquitous Computing*, 13:77–83, January 2009.

[65] G. B. Meisner. "Phi - The Golden Number. The Human Face". Retrieved August, 2009, from `http://www.goldennumber.net/face.htm`.

[66] Z. Mo, J.P. Lewis, and U. Neumann. Improved automatic caricature by feature normalization and exaggeration. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Sketches*, page 57, NY, USA, 2004. ACM.

[67] R. Mukundan and K.R. Ramakrishnan. *Moment Functions in Image Analysis - Theory and Applications*. World Scientific Publishing Co. Pte Ltd, Singapore, 1998.

[68] J.Y. Noh and U. Neumann. A survey of facial modeling and animation techniques. Technical Report 99-705, University of Southern California, 1998.

[69] M. Obaid. *Moment Based Painterly Rendering Using Connected Color Components*. Masters thesis, University of Canterbury, Department of Computer Science and Software Engineering, 2006.

[70] I. S. Pandzic. Facial motion cloning. *Graphical Models*, 65(6):385 – 404, 2003.

[71] I.S. Pandzic. Facial animation framework for the web and mobile platforms. In *Web3D '02: Proceedings of the Seventh International Conference on 3D Web Technology*, pages 27–34, New York, NY, USA, 2002. ACM.

[72] I.S. Pandzic and R. Forchheimer, editors. *MPEG-4 Facial Animation: The Standard, Implementation and Applications*. John Wiley & Sons, Inc., NY, USA, 2003.

[73] A. Paradiso. An algebra for combining mpeg-4 facial animations. In *Proceedings of the Workshop on Life-like Characters: Tools, Affective Functions and Applications*, August 2002.

[74] A. Paradiso. An algebra of facial expressions. In *the proceedings of the First International Joint Conference on Autonomous Agnets and Multi-agent Systems*, July 2002.

[75] F.I. Parke. Computer generated animation of faces. In *ACM '72: Proceedings of the ACM Annual Conference*, pages 451–457, New York, NY, USA, 1972. ACM.

[76] F.I. Parke. Parameterized models for facial animation. *IEEE Computer Graphics and Applications*, 2(9):61–68, 1982.

[77] F.I. Parke and K. Waters. *Computer Facial Animation*. AK Peters Ltd, 2008.

[78] S. Pasquariello and C. Pelachaud. Greta: A simple facial animation

engine. In *Proceedings of the 6th Online World Conference on Soft Computing in Industrial Applications*, 2001.

[79] S. Pasquariello and C. Pelachaud. Greta: A simple facial animation engine. In *Proceedings of the 6th Online World Conference on Soft Computing in Industrial Applications*, 2001.

[80] S.M. Platt and N.I. Badler. Animating facial expressions. *SIGGRAPH Computer Graphics*, 15(3):245–252, 1981.

[81] R. Plutchik. *Emotion: A Psychoevolutionary Synthesis*. Harper and Row, 1980.

[82] M . Radovan and L. Pretorius. Facial animation in a nutshell: past, present and future. In *SAICSIT '06: Proceedings of the 2006 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on IT Research in Developing Countries*, pages 71–79, Republic of South Africa, 2006. South African Institute for Computer Scientists and Information Technologists.

[83] A. Raouzaiou, K. Karpouzis, and S. Kollias. Emotion synthesis in virtual environments. In *Proceedings of 6th International Conference on Enterprise Information Systems*, 2004.

[84] A. Raouzaiou, N. Tsapatsoulis, K. Karpouzis, and S. Kollias. Parameterized facial expression synthesis based on mpeg-4. *EURASIP Journal on Applied Signal Processing*, 2002(1):1021–1038, 2002.

[85] S.S. Reddi. Radial and angular moment invariants for image identification. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 3, pages 240–242, 1981.

[86] L. Redman. *How to Draw Caricatures*. Contemporary Books, Chicago, 1984.

[87] J. Saragih and R. Göcke. Learning AAM fitting through simulation. *Pattern Recognition*, 42(11):2628–2636, 2009. doi: 10.1016/j.patcog.2009.04.014.

[88] J. Saragih and R. Goecke. Iterative error bound minimisation for AAM alignment. In *International Conference on Pattern Recognition ICPR2006*, volume 2, pages 1192–1195, Hong Kong, 2006. IEEE Computer Society. doi: 10.1109/ICPR.2006.730.

[89] J. Saragih and R. Goecke. A nonlinear discriminative approach to AAM fitting. In *Proceedings of the IEEE International Conference on Computer Vision ICCV2007*, Rio de Janeiro, Brazil, 2007. IEEE Computer Society. doi: 10.1109/ICCV.2007.4409106.

[90] T.W. Sederberg and S.R. Parry. Free-form deformation of solid geometric models. *SIGGRAPH Computer Graphics*, 20(4):151–160, 1986.

[91] E. Shimizu and T. Fuse. Rubber-sheeting of historical maps in gis and its application to landscape visualization old-time cities: focusing on tokyo of the past. In CD-ROM, editor, *Proceedings of the 8th In-*

ternational Conference on Computers in Urban Planning and Urban Management, 2003.

[92] M. Shiraishi and Y. Yamaguchi. An algorithm for automatic painterly rendering based on local source image approximation. In *NPAR '00: Proceedings of the 1st International Symposium on Non-Photorealistic Animation and Rendering*, pages 53–58, New York, NY, USA, 2000. ACM.

[93] E. Sifakis, I. Neverov, and R. Fedkiw. Automatic determination of facial muscle activations from sparse motion capture marker data. *ACM Transactions on Graphics - Proceedings of ACM SIGGRAPH 2005*, 24(3):417–425, 2005.

[94] M. Sindeyev and V. Konushin. A novel interactive image matting framework. *GraphiCon*, pages 41–45, 2008.

[95] F.W. Smith and M.H. Wright. Automatic ship photo interpretation by the method of moments. In *IEEE Transactions on Computers*, pages vol. C–26, 1089–1094, 1977.

[96] Y. Su, Y. Liu, U. Zhu, , and Z. Ren. Facial sketch rendering and animation for fun communications. *Interactive technologies and sociotechnical systems: 12th International Conference, VSMM*, 1:486–494, 2006.

[97] M.R. Teague. Image analysis via the general theory of moments. In *Journal of Optical Society of America*, volume 70, pages 920–930, 1980.

[98] A.M. Tekalp and J. Ostermann. Face and 2-d mesh animation in mpeg-4. *Signal Processing: Image Communication*, 15(4-5):387 – 421, 2000.

[99] J. Teran, E. Sifakis, S.S. Blemker, V. Ng-Thow-Hing, C. Lau, and R. Fedkiw. Creating and simulating skeletal muscle from the visible human data set. *IEEE Transactions on Visualization and Computer Graphics*, 11(3):317–328, 2005.

[100] D. Terzopoulos and K. Waters. Physically-based facial modeling, analysis, and animation. *Journal of Visualization and Computer Animation*, 1:73–80, 1990.

[101] C. Tseng and J. Lien. Synthesis of exaggerative caricature with inter and intra correlations. In *8th Asian Conference on Computer Vision*, pages 314–323, Tokyo, Japan, 2007.

[102] Carnegie Mellon University. Rebotic Institute, Automated Face Analysis Group. "FACS - Facial Action Coding System". Retrieved September, 2009, from `http://www.cs.cmu.edu/~face/index2.htm`.

[103] M.L. Viaud and H. Yahia. Facial animation with wrinkles. Research Report RR-1753, INRIA, 1992. Projet SYNTIM.

[104] V. Vinayagamoorthy, M. Gillies, A. Steed, E. Tanguy, X. Pan, C. Loscos, and M. Slater. Building expression into virtual characters. In *Eurographics Conference State of the Art Reports*, 2006.

[105] K. Water. A muscle model for animating three-dimensional facial expression. In *SIGGRAPH '87: Proceedings of the 14th Annual Confer-*

ence on Computer Graphics and Interactive Techniques, pages 17–24, NY, USA, 1987. ACM.

[106] K. Waters and T.M. Levergood. Decface: An automatic lip-synchronization algorithm for synthetic faces. Technical report, Multimedia Tools and Applications, 1993.

[107] E.W. Weisstein. Least squares fitting–polynomial. *MathWorld–A Wolfram Web Resource*, August 2008. `http://mathworld.wolfram.com/LeastSquaresFittingPolynomial.html`.

[108] E.W. Weisstein. "Affine transformation.". *From MathWorld–A Wolfram Web Resource*, August 2010. `http://mathworld.wolfram.com/AffineTransformation.html`.

[109] C.M. Whissel. The dictionary of affect in language. In R. Plutchik and H. Kellerman, editors, *Emotion: Theory, Research and Experience*, volume 4, pages 113–131. Academic Press, 1989.

[110] Wikipedia. "Facial Expression". Retrieved July, 2010, from `http://en.wikipedia.org/wiki/Facial_expression`.

[111] M. Wimmer, U. Zucker, and B. Radig. Human capabilities on video-based facial expression recognition. pages 7–10, 2007.

[112] W. Wong, W. Siu, and K. Lam. Generation of moment invariants and their uses for character recognition. *Pattern Recognition Letters*, 16(2):115–123, 1995.

[113] W.H. Wong and W.C. Siu. Improved digital filter structure for fast moments computation. *IEEE Proceedings - Vision, Image, and Signal Processing*, 146(2):73–79, 1995.

[114] K. Zakharov. Affect recognition and support in intelligent tutoring systems. Masters thesis, University of Canterbury, New Zealand, 2007.

[115] U. Zucker. Facial Expression Recognition- A Comparison between Humans and Algorithms. Technical University of Munich, 2008. `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.85.3342`.